

NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD-A268 525



DTIC
ELECTE
AUG 25 1993
S B D

THESIS

**HEAT TRANSFER PARAMETRIC
SYSTEM IDENTIFICATION**

by

Gregory K. Parker

June, 1993

Thesis Advisor:

Morris R. Driels

Approved for public release, distribution is unlimited.

83 8 24 037

93-19701



1168

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302 and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 1993	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE Heat Transfer Parametric System Identification			5. FUNDING NUMBERS	
6. AUTHOR(S) Parker, Gregory K.				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, California 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This work describes the development of a computer based model that would allow for determination of the transient thermal response characteristics of jet vanes of any size. The model used a thermal lump approach method, considering only conductive and convective heat transfer properties. A constrained optimizer was used to adjust the unknown variables until an adequate match was achieved between the calculated values of the energy balance equations and the experimental data obtained from test firings of a rocket motor. The full scale modeling results were compared to previous quarter scale models in an attempt to determine the applicability of the quarter scale results to full scale vanes. It was determined that the quarter scale model did not provide an accurate representation of the heat transfer process in larger scale vanes, although the full scale model provided a sufficient representation of the thermal response of the jet vane.				
14. SUBJECT TERMS Thrust Vector Control Jet Vane, TVC, STARS; Full Scale Modeling Jet Vane; Constrained Optimizer			15. NUMBER OF PAGES 118	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

Approved for public release; distribution is unlimited.

Heat Transfer Parametric
System Identification

by

Gregory K. Parker
Lieutenant, United States Navy
B.S., DeVry Institute of Technology, 1984

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

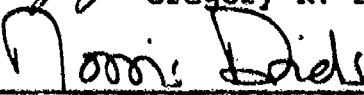
June 1993

Author:



Gregory K. Parker

Approved by:



Morris R. Driels, Thesis Advisor



Matthew D. Kelleher, Chairman
Department of Mechanical Engineering

ABSTRACT

This work describes the development of a computer based model that would allow for determination of the transient response characteristics of jet vanes of any size. The model used a thermal lump approach method, considering only conductive and convective heat transfer properties. A constrained optimizer was used to adjust the unknown variables until an adequate match was achieved between the calculated values of the energy balance equations and the experimental data obtained from test firings of a rocket motor. The full scale modeling results were compared to previous quarter scale models in an attempt to determine the applicability of the quarter scale results to full scale vanes. It was determined that the quarter scale model did not provide an accurate representation of the heat transfer process in larger scale vanes, although the full scale model provided a sufficient representation of the thermal response of the jet vane.

DTIC QUALITY INSPECTED 3

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	BACKGROUND	3
A.	LUMPED PARAMETER MODELING	3
1.	Vane Modeling Concept	3
2.	Lumped Parameter Approach	4
3.	Parametric System Identification	4
B.	BASIC MODELING PROCESSES	5
1.	Equation Formulation	5
2.	Previous Models	12
C.	MODEL SCALING	15
1.	Purpose of Quarter Scale Model	15
2.	Quarter to Full Scale Model Predictions	16
3.	Scaling Analysis	17
III.	FULL SCALE MODELING	20
A.	SIX NODE MODEL APPROACH	20
1.	Concept	20
2.	Geometric Modification	22
3.	Energy Balance Equation Formulation	23
4.	Program Formulation	25
5.	Experimental Data Processing	27

6. Forward Model Matching	30
a. Comparison of Steady State Temperatures	30
b. Response Time Characteristics	34
7. Model Simulation ZXSSQ Optimizer	36
a. Unknowns and Assumptions	36
b. Utilization of Experimental Vane Temp Profiles	37
c. Results	37
8. Model Simulation DBCLSF Optimizer	38
a. Use of Constrained Optimizer	38
b. Boundary Limitations	38
c. Results	39
B. FIVE NODE MODEL APPROACH	40
1. Concept Behind Modification From Six Node Model	40
2. Energy Balance Equation Formulation	42
3. Program Modification	43
4. Model Simulation ZXSSQ Optimizer	44
a. Unknowns and Assumptions	44
b. Experimental Data Used	45
c. Results	45
5. Model Simulation DBCLSF Optimizer	46
a. Constrained Model Simulation	46
b. Single Node Temperature Profile Modeling	46
(1) Node Two Temperature Modeling.	47
(2) Node Three Temperature Modeling.	48

(3) Node Four Temperature Modeling. . .	49
(4) Results.	50
c. Manual Adjustment of Variables-Forward Model	51
d. Two Node Temperature Profile Modeling .	55
(1) Nodes Two and Three Temperature Modeling.	55
(2) Nodes Two and Four Temperature Modeling.	56
(3) Nodes Three and Four Temperature Modeling.	57
(4) Results.	58
e. Convergent Simulation Model	59
(1) Three Node Temperature Profile Matching.	59
(2) Unknowns and Assumptions.	59
(3) Results.	60
f. Simulation With Various Initial Conditions	62
g. Calculations of Heat Transfer Coefficients	63
DISCUSSION OF RESULTS	65
CONCLUSIONS	73

I. INTRODUCTION

The work reported in this document is further research in the study of Thrust Vector Control (TVC) system used in missile trajectory guidance during initial phases of a missiles flight [Ref. 1]. TVC utilizes a system which deflects exhaust thrust to control the missile during launch and initial flight stages. For this phase of flight the vane is actuated from a stowed position and inserted into the exhaust of the rocket nozzle to provide the required control. Figure 1.1 illustrates the stowed and active positions of the TVC vane.

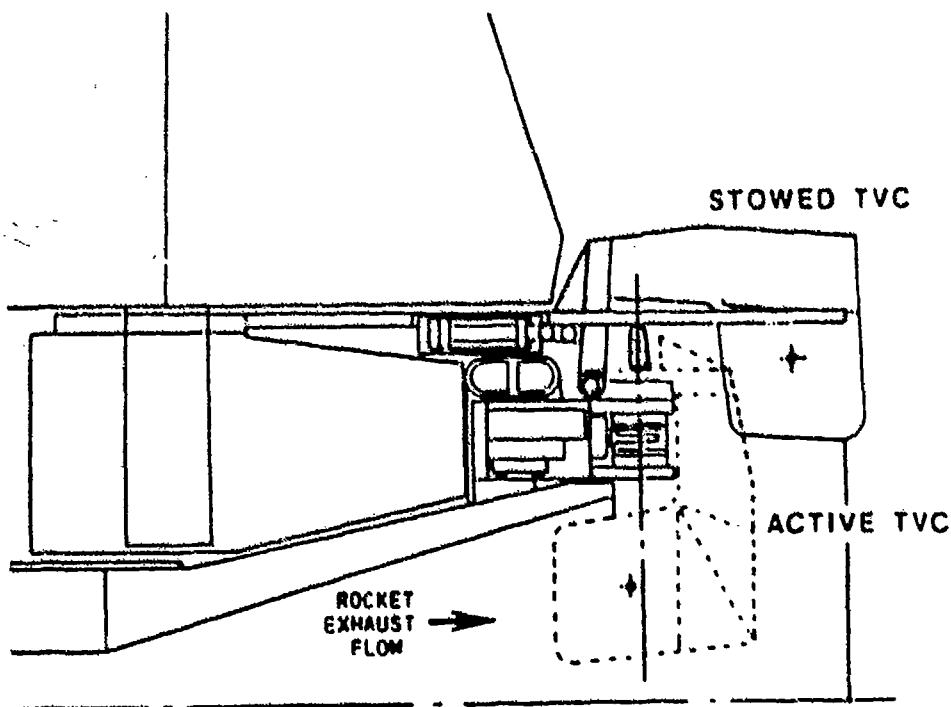


Figure 1.1 STARS TVC Stowage Concept

The vane operates in an extremely harsh thermal environment. In order to use adequate materials for vane design the heat transfer characteristics of a selected material operating in such an environment needs to be resolved. In order to determine the best performance characteristics of different materials a mathematical model needs to be constructed. The model needs to contain both the known physical properties of the material and the unknown properties to be determined. It will utilize the experimental temperature time histories from rocket firings and adjust the unknowns physical properties until the predicted temperature time history and the experimental data are close in a least squares sense.

Past work in the system identification process has been conducted using a quarter scale model to determine the convective heat transfer coefficients and to verify the applicability of the quarter scale results in predicting the heat transfer characteristics of a full scale prototype. The idea behind this report is to enlarge the model to full scale in order to obtain the heat transfer properties and to compare these results with those obtained from the quarter scale modeling process.

II. BACKGROUND

A. LUMPED PARAMETER MODELING

1. Vane Modeling Concept

The jet vane that is part of the TVC system concept will be used as a deflection device to direct the thrust of the rocket exhaust. The vane will be encountering effects from such things as complex thermal changes, supersonic particle laden stream flow, shock waves impinging on the vane and phase variations resulting from vane ablation [Ref. 2:p. 5]. The idea of vane modeling is a process by which some or all of these processes may be modeled in order to find the material that provides the durability and the necessary performance characteristics. Due to the mathematical complexity and the wide number of variables that may effect vane performance it is necessary to limit the model to a smaller scale. The main emphasis of this report is to limit the model to determining the thermal convective characteristics. This modeling concept involves development of a thermal model taking into account the conductive and convective effects of the material selected. Radiation effects will not be considered in the development of this model. The model uses the concept that the transient thermal response of the vane can be developed into a set of equations

that describe the heat transfer process occurring over a distinct period of time.

2. Lumped Parameter Approach

The thermal modeling approach taken will be to break the vane down into distinct thermal lumps [Ref. 2:p. 10]. This approach lumps the heat transfer parameters into distinct nodes. The geometric data and physical properties of the vane material at are then used to determine the conductive resistance and capacitances at each node. The convective resistances between the exhaust plume and the material are essentially unknown and will be determined using the modeling process. The lumped parameter approach is essentially a division of the vane into critical locations, usually chosen to correspond to a thermocouple attachment point on the experimental model, in order to simplify the modeling process [Ref. 2:p. 12]. This simplification enables the formulation of energy balance equations, mentioned previously, that describe the transient thermal response of the individual nodes in the vane [Ref. 1:p. 4]. The energy balance equations take into account thermal energy into and out of the vane nodes and the temperature of the vane accounts for the energy stored at the nodes.

3. Parametric System Identification

The thermal model will be computer-based FORTRAN code using Parametric System Identification (PSI) [Ref. 3:p. 2].

PSI is a procedure in which the model parameters generate a temperature time history that predicts the experimental temperature time history [Ref. 2:p. 6]. The process involves adjustment of the unknown parameters until a desired agreement is reached between experimental and model temperature profiles. The PSI process involves adjustment of the unknown values for the convective coefficients until the model temperature time histories of selected nodes are matched with the experimental temperature time histories of the corresponding nodes on the actual vane. The PSI process brings together the mathematical model and the experimental data. The accuracy of the match between model and experimental data is limited to the complexity of the energy equations used to describe the model itself. The process can be a self adjusting one in that the model can be used to design the vane for experimental testing and then the resulting experimental data can be used to update the original model [Ref. 2:p. 6].

B. BASIC MODELING PROCESSES

1. Equation Formulation

The modeling of the TVC vane involves generating the energy balance equations for the lumped parameter model. The energy equations are formulated by considering the flow of energy into and out of a node. Figure 2.1 illustrates the energy flow concept of a vane node.

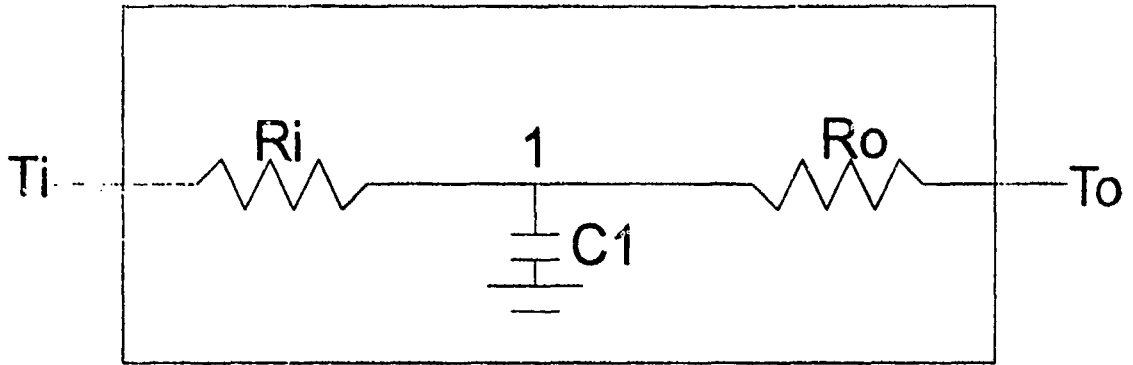


Figure 2.1 Thermal Node Energy Flow

The energy balance formulation for Figure 2.1 can be constructed by considering the flow of energy into and out of the node. This energy balance is represented by,

$$\dot{T}_1 = \frac{T_I}{C_1 R_I} - \frac{T_1}{C_1 R_I} - \frac{T_1}{C_1 R_O} + \frac{T_O}{C_1 R_O} \quad (1)$$

where the value of $T_I > T_1 > T_O$. Starting with the determination of the resistances and capacitances at each individual node from the geometric data and the physical properties of the material is required. Prior to understanding the effects of sub-scale model the conductive and convective parameters first need to be addressed. The conductive resistance is found by,

$$R = \frac{L}{KA_x} \quad (2)$$

where L is the conductive path length, A_x is the cross sectional area of the conductive path and K is the thermal

conductivity [Ref. 3:p. 7]. Likewise, the convective resistance can be found by,

$$R = \frac{1}{hA_s} \quad (3)$$

where A_s is the convective surface area and h is the convective heat transfer coefficient [Ref. 3:p. 7]. The thermal capacitance is found by,

$$C = \rho C_p V \quad (4)$$

where ρ is the material density, C_p is the specific heat and V is the volume [Ref. 3:p. 7]. It can be seen that these values are dependent on not only the material in which the vane is constructed but also the actual size of the vane being modelled. All geometric data is given in full scale and the physical properties are assumed constant no matter what size model is being considered.

The vane model is generated by formulating the energy balance equations for each individual node of the vane. The equations will be dependent on the number of nodes used to model the heat transfer process occurring in the vane. To develop such a set of equations the nodes locations need to be established on the vane. The nodes chosen should correspond to a center of mass or the major sections of the object being modeled. In this modeling process the vane is broken down into four distinct sections. Figure 2.2 illustrates a vane with a four node configuration [Ref. 1:p. 6].



In Figure 2.2 the vane is broken down into four distinct nodes. Node one is for modeling the tip, node two for the fin, node three for the shaft and node four for modeling the mount [Ref. 1:p. 5]. The energy balance equations are generated by considering the heat transfer into and out of each individual node. Nodes one and two, two and three, three and four, are connected by conductive paths with a conductive

resistance linking the nodes [Ref. 3:p. 4]. The designation R_{ij} is used to denote the conductive resistance between nodes i and j . Nodes one and two are linked convectively to two driving temperatures $TR1$ and $TR2$. These are known as the recovery temperatures and are the source of the heating process [Ref. 3:p. 4]. $TR1$ corresponds to the stagnation flow while $TR2$ represents the temperature of the free stream flow. The convective resistances are represented by the designation R_{Fi} , where the i represents the node that the convective temperature is driving. Each node contains a thermal capacitance represented by C_i where i is the node representation. The generation of the energy balance equations involves looking at each individual node and applying the law of conservation of energy [Ref. 3:p. 4]. The conservation of energy for each node involves energy transfer in and out of the node. That is the rate of change of energy at a node is equal to the energy into the node minus the energy out of the node. The remaining energy at the node is what causes the temperature change at that particular location. The node is a point location and the transient thermal heating is being considered. Application of this idea leads to the following equations [Ref. 3:p. 4]:

$$\dot{T}_1 = -\frac{T_1}{C_1 R_{F1}} - \frac{T_1}{C_1 R_{12}} + \frac{T_2}{C_1 R_{12}} + \frac{T_{F1}}{C_1 R_{F1}} \quad (5)$$

$$\dot{T}_2 = \frac{T_1}{C_2 R_{12}} - \frac{T_2}{C_2 R_{f2}} - \frac{T_2}{C_2 R_{12}} - \frac{T_2}{C_2 R_{23}} + \frac{T_3}{C_2 R_{23}} + \frac{T_{R2}}{C_2 R_{f2}} \quad (6)$$

$$\dot{T}_3 = \frac{T_2}{C_3 R_{23}} - \frac{T_3}{C_3 R_{23}} - \frac{T_3}{C_3 R_{34}} + \frac{T_4}{C_3 R_{34}} \quad (7)$$

$$\dot{T}_4 = \frac{T_3}{C_4 R_{34}} - \frac{T_4}{C_4 R_{34}} - \frac{T_4}{C_4 R_{40}} \quad (8)$$

In order to make computer coding easier a simplification of the above equations is desired. The following characteristic rate coefficients are defined [Ref. 3:p. 9]:

$$a_{12} = \frac{1}{C_1 R_{12}} \quad a_{21} = \frac{1}{C_2 R_{12}} \quad a_{23} = \frac{1}{C_2 R_{23}} \quad (9)$$

$$a_{32} = \frac{1}{C_3 R_{23}} \quad a_{34} = \frac{1}{C_3 R_{34}} \quad a_{43} = \frac{1}{C_4 R_{34}} \quad (10)$$

$$a_{40} = \frac{1}{C_4 R_{40}} \quad b_{11} = \frac{1}{C_1 R_{f1}} \quad b_{22} = \frac{1}{C_2 R_{f2}} \quad (11)$$

A further simplification can be made by combining coefficients for the same temperature at a particular node. This simplification allows each node temperature T_i to have a single coefficient making it possible to put the energy balance equations into matrix form. This is done as follows [Ref. 3:p. 5]; The energy balance equations now become [Ref. 3:p. 5],

$$a_{11}=a_{12}+b_{11} \quad a_{22}=a_{21}+a_{23}+b_{22} \quad (12)$$

$$a_{33}=a_{32}+a_{34} \quad a_{44}=a_{43}+a_{46} \quad (13)$$

$$\dot{T}_1 = -a_{11}T_1 + a_{12}T_2 + b_{11}T_{R1} \quad (14)$$

$$\dot{T}_2 = a_{21}T_1 - a_{22}T_2 + a_{23}T_3 + b_{22}T_{R2} \quad (15)$$

$$\dot{T}_3 = a_{32}T_2 - a_{33}T_3 + a_{34}T_4 \quad (16)$$

$$\dot{T}_4 = a_{43}T_3 - a_{44}T_4 \quad (17)$$

These equations may be put in matrix form to give a clearer understanding of there use in the computer coding
[Ref. 3:p. 5].

$$\begin{bmatrix} \dot{T}_1 \\ \dot{T}_2 \\ \dot{T}_3 \\ \dot{T}_4 \end{bmatrix} = \begin{bmatrix} -a_{11} & a_{12} & 0 & 0 \\ a_{12} & -a_{22} & a_{23} & 0 \\ 0 & a_{32} & -a_{33} & a_{34} \\ 0 & 0 & a_{43} & -a_{44} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} + \begin{bmatrix} b_{11} & 0 & 0 & 0 \\ 0 & b_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T_{R1} \\ T_{R2} \\ 0 \\ 0 \end{bmatrix} \quad (18)$$

This is a compact notation commonly known as a state space equation and may be written as,

$$\dot{x} = Ax + Bu \quad (19)$$

Equations of this form are common and a wide variety of package are available for solving linear differential equations.

2. Previous Models

Work on the jet vane thermal model was started by Nunn and Kelleher in 1986 at the Naval Postgraduate School [Ref. 5]. Further research into the model was conducted by Nunn [Ref. 2], Hatzenbuehler [Ref. 4] , Reno [Ref. 1] and Driels [Ref. 3]. Prior work has concentrated on development of a quarter scale model and its applicability to full scale vanes. Reno used both 3 and 4 node approaches in modeling the heat transfer process of the vane. The experimental data used for the comparison process, on the quarter scale model, was from the shaft and mount locations [Ref. 1:p. 7]. The points were chosen were due to the limited size of the experimental vane for thermocouple attachment. Those points being the back of the shaft and mount. The 4 node vane model in figure 2.1 illustrates the set-up used by Reno along with the energy balance equations. The unknown variables that needed to be determined were R_{F2} , R_{34} , R_{4G} and C_4 [Ref. 1:p. 10]. In the simplified energy balance equations these coincide to a_{34} , a_{43} , a_{4G} and b_{22} . The geometric data used for the full scale vane is given in Table 1 [Ref. 3:p. 8].

TABLE 1

	tip	to	vane	to	shaft
V, cm ³	2.6		52		23
A _x , cm ²		5.9		5.2	
L, cm		5.0		6.0	

The physical data for the vane, which was a tungsten matrix impregnated with copper, is $\rho = 18310 \text{ kg/m}^3$, $K = 173 \text{ W/m}\cdot\text{K}$, and $C_p = 146 \text{ J/kg}\cdot\text{K}$ [Ref. 3:p. 7]. The recovery temperatures used were $TR1 = 2360 \text{ K}$ and $TR2 = 2260 \text{ K}$. The input thrust profile used was simulated as a ramp function. The recovery temperatures, $TR1$ and $TR2$, contained in the input vector u , of the matrix form, drive the system, and are the product of the steady state values and the thrust function. Figure 2.3 illustrates this functional input [Ref. 3].

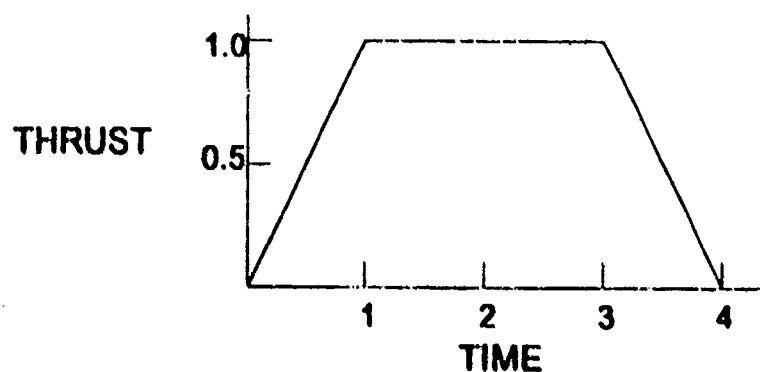


Figure 2.3 Ramp Function for Recovery Temperatures

Reno utilized a personal computer with a program System Build and System Identification developed by Integrated Systems, Inc. [Ref. 1:p. 10]. The identification process used MAXLIKE which is a function of MATRIX X [Ref. 1:p. 10]. This program allowed the four unknown parameters to vary until the temperature profiles generated matched those obtained from the experimental firings. The following results were obtained:
 $b_{22} = 0.2718 \text{ s}^{-1}$, $a_{34} = 0.299 \text{ s}^{-1}$, $a_{43} = 0.1322 \text{ s}^{-1}$ and

$a_{4G} = 0.1018 \text{ s}^{-1}$ [Ref. 1:p.10]. These parameters correspond to the values of [Ref. 1:p. 11]: $R_{F2} = 1.67 \text{ K/W}$, $R_{34} = 3.33 \text{ K/W}$, $R_{4G} = 4.33 \text{ K/W}$, $C_4 = 2.27 \text{ J/K}$. There was a good match between the generated and experimental temperature time histories of the mount and the shaft. Figure 2.4 demonstrates the close proximity between the shaft and mount profiles [Ref. 1:p. 12].

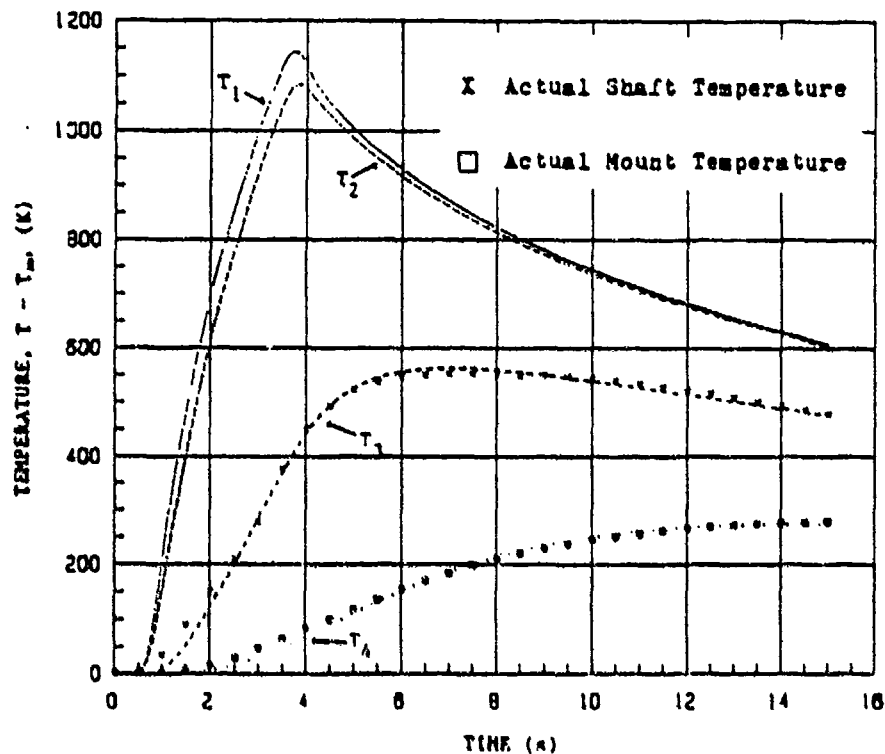


Figure 2.4 Quarter Scale Temperature Profiles

Determination of the unknown parameters is based upon a temperature profile match between the mount and shaft. The temperature profiles temperatures T_1 and T_2 are the mathematical equation profiles and have no experimental data to be matched to in the quarter scale modeling process.

Work done by both Reno [Ref. 1] and Driels [Ref. 3] indicated that although the convective heat transfer coefficient at the tip node was an unknown variable, in the quarter scale modeling process, it had minimal effect on the outcome of the previously identified unknown values and thus b_{11} was held constant. Table 2 illustrates the variation in the four unknowns for various values of b_{11} [Ref. 3:p. 12].

TABLE 2

R_{F1}	a_{34}	a_{43}	a_{4G}	b_{22}
5.35	0.483	0.152	-0.167	0.143
10.0	0.447	0.152	-0.168	0.162
20.0	0.44	0.152	-0.169	0.174
1.0	0.51	0.153	-0.164	0.056
0.5	0.47	0.153	-0.163	0.019

The results reported in Table 2 are those obtained by Driels [Ref. 3] and do not exactly match the results obtained by Reno [Ref. 1]. The temperature profiles obtained by Driels coincide with those displayed in Figure 2.4.

C. MODEL SCALING

1. Purpose of Quarter Scale Model

The quarter scale model was developed to predict the behavior of the heat transfer parameters of a full scale prototype. The quarter scale model quantitatively provides for testing on a smaller scale which in the long term is more cost effective than constructing one of full scale. This

savings come from the reduced size of the rocket motor, the reduced use of data acquisition equipment, environmental requirements that have to be met when firing full scale that do not apply to the quarter scale, manufacturing of quarter scale vanes, material requirements, etc. The quarter scale modeling process, if adequate, could provide a quick, simple form of modeling for a larger vane without the added difficulties.

2. Quarter to Full Scale Model Predictions

It has been demonstrated that a sub-scale model can be generated and subsequent experimental data obtained in order to predict the heat transfer parameters of a given material. A set of convective heat transfer coefficients can be determined from the results of such models. The accuracy of these results and there applicability to a full scale vane has yet to be determined. Direct scaling of the geometric properties is achieved and the physical properties such as density, thermal conductivity and specific are assumed constant for both sub-scale and full scale vanes. The difficulty arises in the effects of the jet plume on the different sizes of vane. As mentioned earlier the conductive properties rely on both physical and geometric data. Variation in the geometric size would have a profound impact on the transient heat transfer process occurring not only inside the vane but due to the larger surface area the effects

of the convective heat transfer process. A direct scaling up of the quarter scale model to full scale may be inadequate to properly model the effects of a large vane. Increases in the rocket chamber temperature thus an increased recovery temperature may also influence the energy transfer process. The purpose of this report is to develop a full scale model, compare the results to those obtained by the quarter scale and determine if the results obtained from the sub-scale tests reflect the actual heat transfer process of a full scale vane.

3. Scaling Analysis

To scale the resistances and capacitances a reduction in the linear dimensions of the vane must be calculated in order that the model reflects the actual size of the manufactured vane in the experimental phase [Ref. 3:p. 8]. A scale factor SF, where $SF < 1$, is introduced to reduce the geometric dimensions of the vane and subsequently altering the resistances and capacitances [Ref. 3:p. 8]. The scale values can be determined by manipulating equations (2), (3) and (4). When manipulating these equations the physical properties such as the thermal conductance k , the specific heat C_p , and the density ρ remain constant and do not change. The equation manipulation results in,

$$R_f = \frac{L}{KA_x} \quad \rightarrow \quad R_s = \frac{L \cdot SF}{KA_x \cdot SF^2} \quad \rightarrow \quad R_s = \frac{L}{KA_x} \frac{1}{SF} \quad (20)$$

$$R_s = \frac{R_f}{SF} \quad (21)$$

where R_f is the full scale resistance and R_s is the sub-scale resistance. An important point to note is that the scale factor SF being less than one causes the resistance to increase during the scaling process. An example of this is the quarter scale models previously constructed where the scaled resistance R_s is equal to the full scale divided by 0.25 (1/4 scale modeling). This results in the scaled resistance being four times the full scale value. The thermal capacitance has the same linear dimension scaling as the resistance did, which is given by,

$$C_f = \rho C_p V = C_s = \rho C_p V SF^3 \quad (22)$$

$$C_s = C_f SF^3 \quad (23)$$

where C_f is the full scale capacitance and as indicated C_s is the sub-scale capacitance [Ref. 3:p. 8]. There is no direct scaling of the convective resistance in the model due to the fact that the heat transfer coefficient h is unknown. The only scaling used on the convective values is if convergence has been achieved and the value of the convective resistance is needed. The scaling process for the convective resistance

is similar to those previously stated and is given by [Ref. 3:p. 8],

$$P_s = \frac{R_f}{SF^2} \quad (24)$$

Although there is an increase in the resistance due to scaling and a decrease in the capacitance, the combined effect of these two values must be emphasized. The product of RC is a time constant for a particular driving temperature [Ref. 1:p. 7]. Scaling of the values of R and C results in a change in the time constants for the energy balance equations from the full scale to the sub-scale. The resultant change is demonstrated in the following formula;

$$R_s C_s = \frac{R_f}{SF} C_f \cdot SF^3 \Rightarrow R_s C_s = R_f C_f \cdot SF^2 \quad (25)$$

The time constant for the sub-scale values decreases as the product of the full scale time constant times the square of the scale factor. It will be seen that in formulating the energy balance equations for the mathematical model that the scaling of the time constants will have a profound effect on the conductive and convective values.

III. FULL SCALE MODELING

A. SIX NODE MODEL APPROACH

1. Concept

As discussed previously, prior quarter scale modeling approaches involved using three and four node models. Placement of the node points in the model were dependent on the center of mass at determined critical locations and the availability of experimental data from a quarter scale test firing. The experimental data coming from the shaft and mount locations. These temperature profiles were matched by the mathematical model and the results used in determining the convective heat transfer coefficient for the fin section, the unknown resistance between the mount and shaft, and the unknown capacitance of the mount. A single value of convective heat transfer coefficient was obtained from the quarter scale tests but due to the lack of experimental temperature profiles from that particular location the applicability of this value to a larger vane was in question.

The basis for increasing the number of nodes when attempting to model the full scale vane was two fold; 1) more node locations would give a more accurate representation of the actual heat transfer process going on in the vane and 2) these added node locations had experimental data available

which allowed for a more diverse set of temperature time profiles that could be used in constructing the mathematical model. The 21 locations that experimental data was taken from allows a greater flexibility in the selection of temperature profiles to be matched. Although Nunn [Ref. 2:p 12] recommended that for simplicity and efficiency the minimum number of nodes should be used, to accurately model the heat transfer process of a full scale vane a larger number of nodes was needed. The modification was made by adding two more node locations to the four node model in the fin area. Figure 3.1 shows the six node model design.

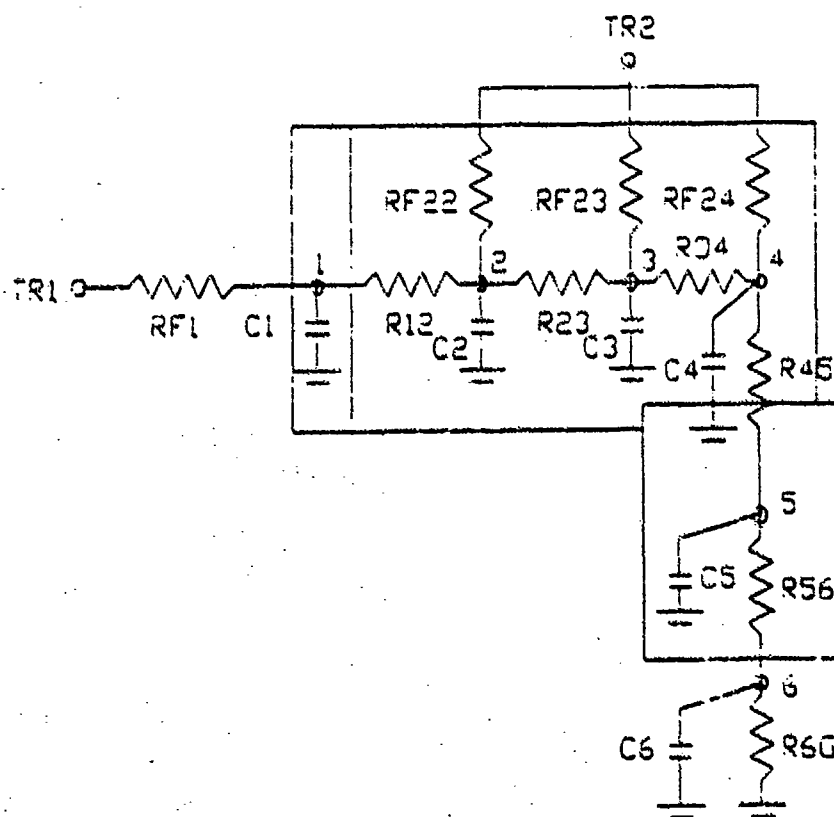


Figure 3.1 Six Node Vane Model

2. Geometric Modification

The addition of two nodes to the fin in the six node model require a modification of the geometric properties in order that the proper volumes, areas and lengths are used in calculating the resistances and capacitances. These new values are given in Table 3,

TABLE 3

	tip	to node2	to node3	to node4	to shaft
V, cm ³	2.6	12.86	17.15	21.44	23.0
A _x , cm ²		5.6	5.9	6.2	5.2
L, cm		2.774	2.774	2.774	5.0

The values are based on those obtained from Table 1 [Ref. 3:p. 8] and the four node model design. The parameters for the node volumes were obtained by taking the vane volume and estimating a 20% change in the volume from the rear of the vane, node four section, to the forward part of the vane, node two section, with the rear being the largest and the basis for the calculations. The cross sectional areas were determined in the same manner with the area at node two being considered to be 20% less than node three and node four cross sectional area to be 20% greater than that of node three. The cross sectional area at node three was known because it corresponds to node two of the quarter scale four node model. The quarter scale model parameters were given in full scale terms and scaled down in the program formulation.

3. Energy Balance Equation Formulation

Formulation of the energy balance equations is done by applying the law of conservation of energy to the six nodes of the vane. Application of this law leads to the following,

$$\dot{T}_1 = -\frac{T_1}{C_1 R_{F1}} - \frac{T_1}{C_1 R_{12}} + \frac{T_2}{C_1 R_{12}} + \frac{T_{R1}}{C_1 R_{F1}} \quad (26)$$

$$\dot{T}_2 = \frac{T_1}{C_2 R_{12}} - \frac{T_2}{C_2 R_{12}} - \frac{T_2}{C_2 R_{F22}} - \frac{T_2}{C_2 R_{23}} + \frac{T_4}{C_2 R_{23}} + \frac{T_{R2}}{C_2 R_{F22}} \quad (27)$$

$$\dot{T}_3 = \frac{T_2}{C_3 R_{23}} - \frac{T_3}{C_3 R_{23}} - \frac{T_3}{C_3 R_{F23}} - \frac{T_3}{C_3 R_{34}} + \frac{T_4}{C_3 R_{34}} + \frac{T_{R2}}{C_3 R_{F23}} \quad (28)$$

$$\dot{T}_4 = \frac{T_3}{C_4 R_{34}} - \frac{T_4}{C_4 R_{34}} - \frac{T_4}{C_4 R_{F24}} - \frac{T_4}{C_4 R_{45}} + \frac{T_5}{C_4 R_{45}} + \frac{T_{R2}}{C_4 R_{F24}} \quad (29)$$

$$\dot{T}_5 = \frac{T_4}{C_5 R_{45}} - \frac{T_5}{C_5 R_{45}} - \frac{T_5}{C_5 R_{56}} + \frac{T_6}{C_5 R_{56}} \quad (30)$$

$$\dot{T}_6 = \frac{T_5}{C_6 R_{56}} - \frac{T_6}{C_6 R_{56}} - \frac{T_6}{C_6 R_{6G}} \quad (31)$$

As discussed in Chapter II this series of equations can be simplified by defining a set of characteristic rate coefficients. These coefficients relate the properties of the vane to the time constants. The simplification involves both a conductive coefficient labeled a_{ij} and a convective coefficient labeled b_{ij} . This simplification leads to,

$$a_{12} = \frac{1}{C_1 R_{12}} \quad a_{21} = \frac{1}{C_2 R_{12}} \quad b_{11} = \frac{1}{C_1 R_{F1}} \quad (32)$$

$$a_{23} = \frac{1}{C_2 R_{23}} \quad a_{32} = \frac{1}{C_3 R_{23}} \quad b_{22} = \frac{1}{C_2 R_{F22}} \quad (33)$$

$$a_{34} = \frac{1}{C_3 R_{34}} \quad a_{43} = \frac{1}{C_4 R_{34}} \quad b_{32} = \frac{1}{C_3 R_{F23}} \quad (34)$$

$$a_{45} = \frac{1}{C_4 R_{45}} \quad a_{54} = \frac{1}{C_5 R_{45}} \quad b_{42} = \frac{1}{C_4 R_{F24}} \quad (35)$$

$$a_{56} = \frac{1}{C_5 R_{56}} \quad a_{65} = \frac{1}{C_6 R_{56}} \quad a_{66} = \frac{1}{C_6 R_{66}} \quad (36)$$

$$a_{11} = b_{11} + a_{12} \quad a_{22} = a_{21} + b_{22} + a_{23} \quad a_{33} = a_{32} + b_{32} + a_{34} \quad (37)$$

$$a_{44} = a_{43} + b_{42} + a_{45} \quad a_{55} = a_{54} + a_{56} \quad a_{66} = a_{65} + a_{66} \quad (38)$$

These simplifications now allow the energy balance equations to be put into matrix form. The modification provides an easier approach to the program formulation. The matrix equation description was described in Chapter II and with the application of the characteristic rate coefficients the lumping of the common temperature coefficients in each energy balance formulation the following equations can be used to

generate the FORTRAN code used to solve these differential equations. The equations become,

$$\dot{T}_1 = -a_{11}T_1 + a_{12}T_2 + b_{11}T_{R1} \quad (39)$$

$$\dot{T}_2 = a_{21}T_1 - a_{22}T_2 + a_{23}T_3 + b_{22}T_{R2} \quad (40)$$

$$\dot{T}_3 = a_{32}T_2 - a_{33}T_3 + a_{34}T_4 + b_{23}T_{R2} \quad (41)$$

$$\dot{T}_4 = a_{43}T_3 - a_{44}T_4 + a_{45}T_5 + b_{24}T_{R2} \quad (42)$$

$$\dot{T}_5 = a_{54}T_4 - a_{55}T_5 + a_{56}T_6 \quad (43)$$

$$\dot{T}_6 = a_{65}T_5 - a_{66}T_6 \quad (44)$$

4. Program Formulation

The energy balance equations are a set of linear ordinary differential equations with some of the rate coefficients as unknowns. The process to evaluate these unknowns is to start with a set of initial conditions for these unknowns, solve the differential equations over a period of time to obtain a temperature time history for each node. We then compare the experimental data from a set of corresponding nodes and the results obtain from the model program and calculate the difference of the sum of the squares between the two temperature profiles. The programs optimizer

adjusts the unknown variables and solves the differential equations again to obtain a new temperature time history. This process is illustrated in Figure 3.2

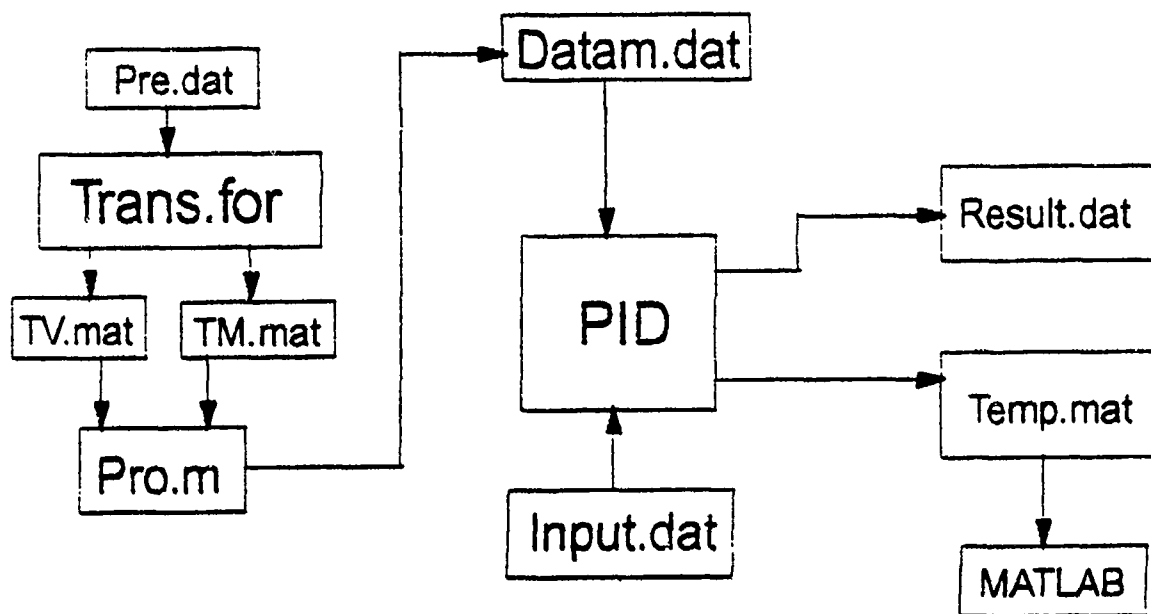


Figure 3.2 Block Diagram of Modeling Program

The blocks labeled Pre.dat, Trans.for, TV.mat, TM.mat and Pro.m will be discussed in the following section. The program PID reads the experimental data from the Datam.dat file. It then reads in the full scale geometric properties of the vane and the physical properties of the material the vane is constructed of from Input.dat. It calculates the resistance and capacitances for the nodes and the rate coefficients a_{ij} . The initial conditions are set for the unknown parameters and PID then calls the optimizer subroutine. The optimizer subroutine intern calls another subroutine called TEMP which calls the needed equation solver and function program that

provides the differential equations. DIVPRK (Double-precision Initial Value Problem Runge-Kutta) is an IMSL routine used to solve linear ordinary differential equations with the given set of initial conditions. The subroutine that supplies the differential equations is called FCN. The values of the driving temperatures are supplied by the main program PID and are common to all the subroutines. The results of DIVPRK are returned to the subroutine TEMP as a temperature time history. TEMP then calculates a sum of the squares error between the experimental data read into the program PID and the results of the equation solver. This error is returned to the optimizer program where the optimizer adjusts the unknown parameters and repeats the process until a set of convergence criteria is met.

Once the program converges the results are written to two files Result.dat and Temp.mat. Results.dat contains the calculated values of the convective resistance and the convective heat transfer coefficient. The temp.mat file contains the experimental and model temperature time histories. This file can be read into MATLAB to be graphed, which gives a visual representation of how close the match is between the two profiles.

5. Experimental Data Processing

The experimental data used in the identification process is obtained from the Naval Weapons Center (NWC), China

Lake, California. The data is received on a computer readout sheet for the different thermocouple locations on the experimental vane. The temperature is measured in degrees Fahrenheit. The computer program needs the temperatures in degrees kelvin and referenced to the ambient or reference temperature. This referencing of the temperature profiles enables the initial conditions of the equation solver to be set to zero. In order to use the experimental data it must first be preprocessed. The preprocessing of the data is illustrated in the block diagram in Figure 3.3.

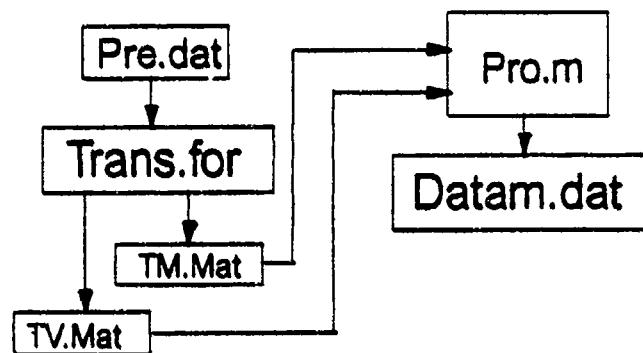


Figure 3.3 Experimental Data Preprocessing Diagram

The Pre.dat file is first produced by manually selecting the series of temperatures of a specified time increment. The separation of the temperatures is up to the individual. The time interval selected when recording the experimental temperatures must also be placed into the TEMP subprogram for the differential Equation solver DIVPRK. This time interval allows the equation solver to calculate the temperature time histories over the same time increments as the experimental

data being used. Failure to match these time increments may cause erroneous values when trying to match experimental and calculated values. The values are entered into the file Pre.dat manually as a nx1 matrix. The raw data from the Pre.dat file is read into a preprocessing program called Trans.for. This smoothing program involves adding the two temperatures prior to the one being averaged, the temperature being averaged and the two temperatures subsequent to that particular temperature. Since there are no temperature values prior to the first two and subsequent to the last two, these values are taken as is. For 61 temperature values over a given period of time temperatures one and two, and 60 and 61 are taken as is. The temperatures 3 through 59 are averaged values once the Trans.for program has been run.

The number of files produced from Trans.for may vary depending on the number of experimental temperature profiles used. The basic files produced are Datam.dat and TV.mat. The TM.mat file is only used if experimental data is being utilized from the mount. The files TV.Mat and TM.mat are read into a MATLAB program called Pro.m. This program normalizes the temperatures to the ambient and converts the temperature from degrees fahrenheit to degress Kelvin. The output of this program is a file called Datam.dat which is a nx1 matrix. Datam.dat is the smoothed file containing the time-temperature profiles of the experimental data. At this point the Datam.dat file is ready to be used by the main program.

6. Forward Model Matching

a. Comparison of Steady State Temperatures

A forward model is the reverse of the vane optimization model. It takes a set of values and generates a set of time-temperature profiles for a given set of data. There is no optimizing of the variables. All the rate coefficients are constants and the profiles generated are based on the solution to the differential equations using these constants over a given period of time. A forward model was constructed from the program PID to check the types of time-temperature profiles that would be generated by these equations given a set of values for the rate coefficients. To construct the forward model from the main program PID the optimizer subroutine was eliminated and the PID program called the TEMP subroutine directly. This forward model allows the user to solve a set of equations over a given period with a set of rate coefficients, a_{ij} and b_{ij} , remaining constant. A set of time-temperature profiles was constructed from the full scale energy balance equations using the final optimized values for the unknowns that Reno found from the quarter scale modeling process [Ref. 1:p. 10]. The values were scaled up from quarter to full scale and inserted into the PID program for the unknown values at the corresponding locations. The values of b_{22} and b_{24} did not correspond to anything that was done by Reno [Ref. 1], so the values for the convective rate

coefficients were all set equal to each other. The forward model was run on the six node vane model for both quarter and full scale using the following values: Quarter Scale - $a_{34}=0.299$, $a_{43}=0.1322$, $a_{46}=0.1018$, $b_{11}=30.163$, $b_{22}=0.271$ Full Scale - $a_{56}=0.01869$, $a_{65}=0.008262$, $a_{66}=0.006362$, $b_{11}=1.885$, $b_{22}=b_{32}=b_{42}=0.017$. If the thrust profile is allowed to continue indefinitely the steady state values of temperature should be the same for both the quarter and full scale values on the six node vane model. These values can be determined analytically by considering a voltage divider across the resistance path to ground. The convective temperature sources act as the supply voltages in such a case. The analytical calculations for both sets come up to the same steady state temperature values for both scaling sizes as shown in Figure 3.4. The main difference is that the quarter scale reached its steady state temperature at a much quicker rate than the full scale. The quarter scale reached steady state in approximately 20 seconds where as the full scale had not reached its steady state temperatures even after 160 seconds. Figure 3.4 shows the quarter scale response of the node temperatures over a 160 seconds firing period.

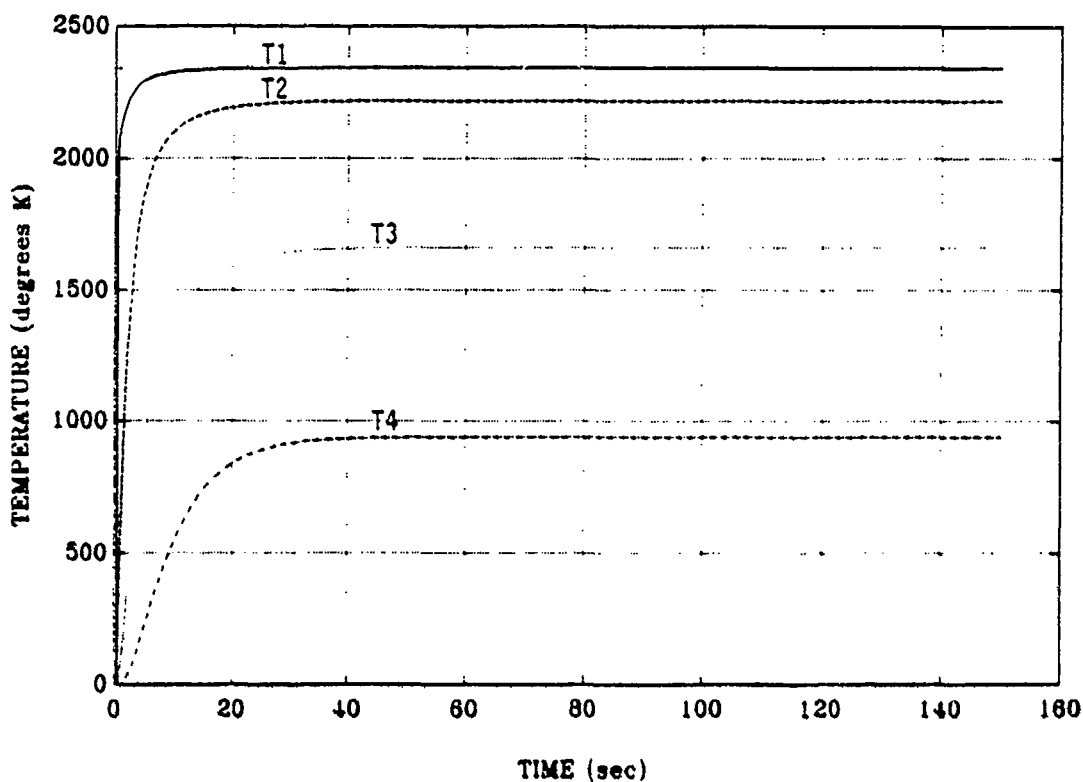


Figure 3.4 Quarter Scale Forward Model Temperature Response

The response curves in Figure 3.4 are for a six node vane model with the nodes at the shaft and mount being neglected. The analytical values of the steady state temperatures at nodes one through four are; $T_1=2340$ K, $T_2=2210$ K, $T_3=1639$ K and $T_4=926$ K, obtained from the figure above. The values of the rate coefficients are set to those corresponding to the full scale vane and the forward model program is executed again. The time temperature profiles for the full scale are listed in Figure 3.5 over the same 160 second firing period.

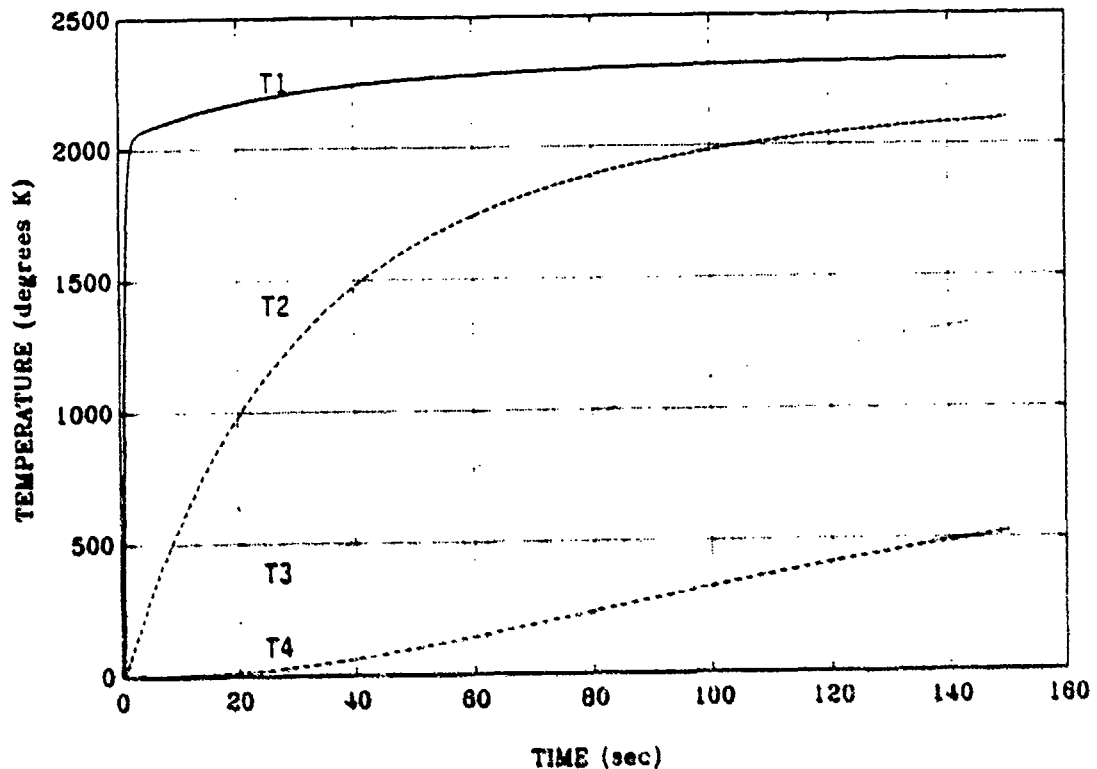


Figure 3.5 Full Scale Forward Model Temperature Response

The analytical calculations yield the same steady state values for nodes one through four. The variation is in the amount of time needed to get to the final value. The full scale requires a much greater period of time to reach the steady state than the quarter scale does, although the full scale will reach the same steady state temperature. This goes back to the time constants of the two different scale models. The rate characteristics decrease for the full scale model but these are the reciprocals of the time constant which actually increase for the full scale model.

b. Response Time Characteristics

A check needs to be made to see if the response characteristics of the energy balance equations for the six node model are similar to those of the four node model obtained from Reno's work [Ref. 1]. The forward model will be utilized with the values of the rate coefficients set equal to those in Reno's thesis [Ref. 1:p. 10]. The thrust profile will be the same as that used by Reno and so will the recovery temperatures, with $T_{R1}=2360$ and $T_{R2}=2260$. The time-temperature profiles obtained by Reno for a four node quarter scale model are shown in Figure 3.6.

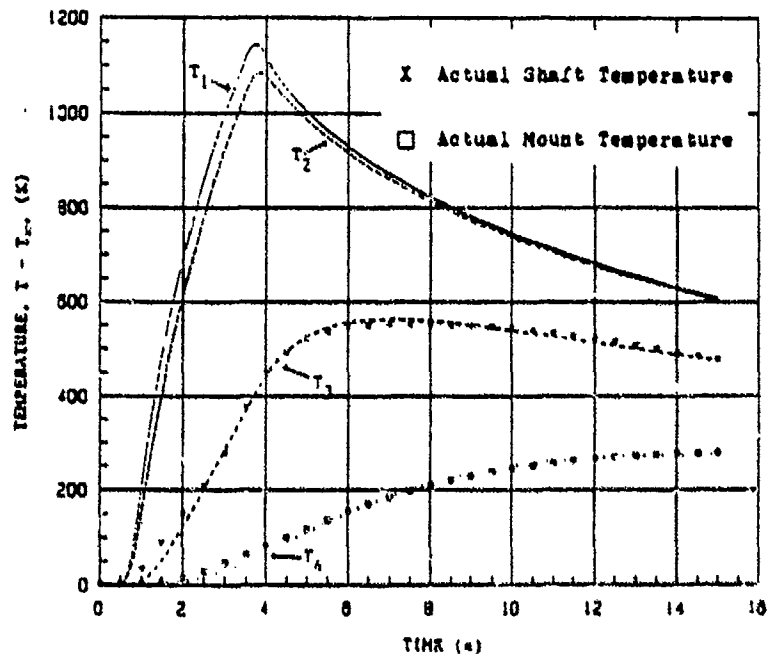


Figure 3.6 Temperature Histories, Four Node Model

The results in Figure 3.6 are based on an optimization program but if the energy balance equations are correct the final values used by Reno should be able to be inserted into the forward model generating a set of temperature profiles similar to those shown above. Figure 3.7 shows the results of this forward model using the six node energy equations with Reno's rate characteristic coefficients.

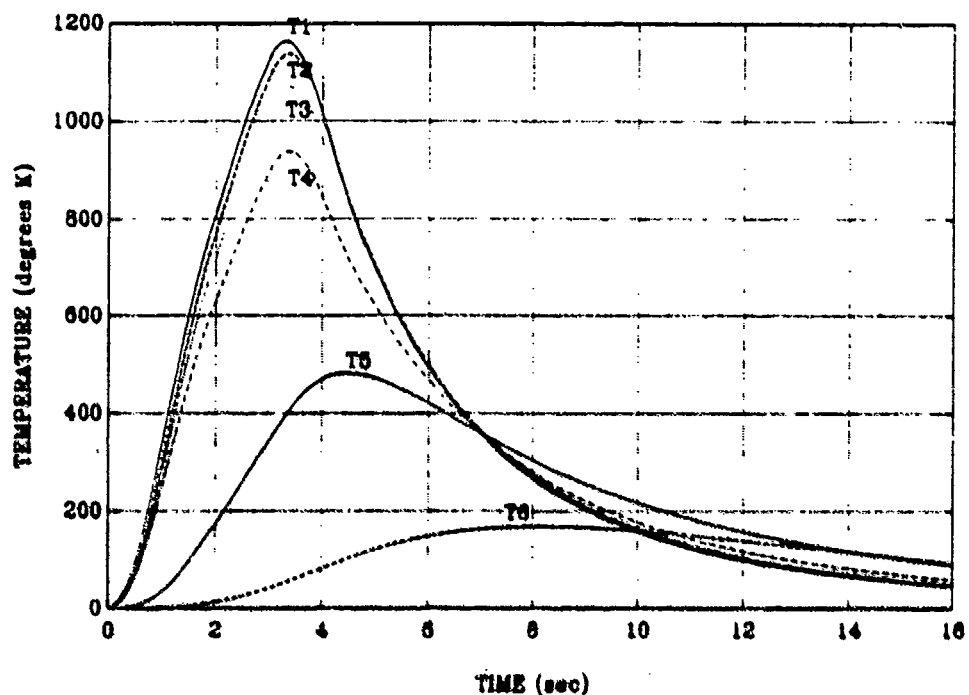


Figure 3.7 Six Node Quarter Scale Forward Model Response
Comparison of the temperature profiles of Figures 3.6 and 3.7 indicate they are very similar with the temperature profiles of T2 from the four node and T3 from the six node corresponding to each other. T3 from the four node corresponds to T5 from the six node and T4 from the four node corresponds to T6 of the six node model.

7. Model Simulation ZXSSQ Optimizer

a. Unknowns and Assumptions

The first series of runs utilized an optimizer known as ZXSSQ which is an old IMSL routine that is stored in a separate library. It must be noted at this point that the thrust profile for the full scale modeling process is not a ramp function as those conducted in the quarter scale process but is being simulated as a step response. The thrust turns on at approximately time zero and shuts off approximately 6.7 seconds later. Prior to starting the PSI process using the program formulation provided, the unknowns must be identified from the energy balance equations and a set of assumptions must be established. The unknown values that need to be determined are the convective resistances from the free stream to the vane, R_{F22} , R_{F23} , and R_{F24} , the conductive resistance between nodes five and six, R_{56} , the conductive resistance from node six to ground, R_{6G} , and the capacitance of node six, C_6 . Although the tip convective resistance R_{F1} and b_{11} are unknown, the first series of the identification process assumed this value to be constant as it was in the quarter scale simulation conducted by Driels [Ref. 3] and Reno [Ref. 1].

Two simulations were conducted; the first with the assumption that the convective resistances to the free stream were dependent and equal, that is $R_{F22}=R_{F23}=R_{F24}$, which makes

$b_{22}=b_{32}=b_{42}$. The second simulation made the assumption that the convective resistances were independent and not equal.

b. Utilization of Experimental Vane Temp Profiles

Prior to model simulation the node locations for the use of experimental data needs to be selected. The quarter scale modeling done in previous work has utilized experimental temperature profiles from the shaft and mount locations. For the full scale vane experimental data is available from the fin area of the vane. A set of simulations will be conducted using the same set of locations as that done by the quarter scale modeling process. Then the experimental data will be changed and the temperature profiles from nodes two, three and four will be utilized as the experimental time-temperature profiles to be matched.

c. Results

A series of simulations were conducted, with the initial conditions set at zero, varying the assumption of dependence and independence of the convective rate coefficients and varying the experimental data used in the identification process. All the simulations had the same outcome, there was no convergence. The values of a_{56} and a_{65} increase substantially in value causing a fatal error three in the differential equation solver giving the error message that the differential equation problems became too stiff. An attempt was made to alter the starting initial conditions in

the hope that the values could be placed closer to their final values. This was a manual attempt to correct the sudden change in the values for the unknowns. Varying the initial conditions had no effect on the outcome. The problem still failed to converge.

8. Model Simulation DBCLSF Optimizer

a. Use of Constrained Optimizer

Since the values of some of the unknowns would jump substantially causing the program to fail, a constrained optimizer was used in an attempt to hold these values within a certain set of limits. The constrained optimizer chosen was DBCLSF which is a current IMSL subroutine. The reasoning behind this was that maybe the jump in the unknowns was caused by a local minima and when the optimizer, ZXSSQ, made a large adjustment it caused the problem to become too stiff. The constrained optimizer would keep the values of the unknowns low enough to prevent the differential equation solver from becoming too stiff and allow it to continue to solve the set of equations given.

b. Boundary Limitations

The constrained optimizer allows the user to select a variety of boundary conditions. This optimizer, DBCLSF, permits the selection of boundary limits that are all negative, all positive, or manually set either to each individual unknown or as a blanket setting for all the

unknowns. A simulation was conducted with the boundary limits set to all positive and then to a set with the lower boundary being set at 0.0001 and the upper boundary set at 100.

c. Results

For the simulation, with the set of initial conditions at zero, there was no convergence with either the boundary limits set all positive or for the upper and lower boundary limits. Again the initial starting values of the unknowns were changed from zero to various values in an attempt to avoid any possible local minima. The results were the same as before the program failed to converge. The program had an arithmetic fault or just continued on indefinitely with the values of the unknowns reaching the boundary limits and remaining there. The program is trying to match a series of time-temperature profiles from the fin area of the vane. The program does not converge due to the substantial jump in the unknowns at the mount location. From the experimental data the temperature profiles at the shaft and mount locations vary little from the ambient. Elimination of the mount node, node six, should have little effect on the predictions for nodes two, three and four. This coupled with the inconsistencies in use of mount configurations leads to the modification of the original six node model to a five node model. A model without a mount node and with the shaft node, node five, going directly to ground.

B. FIVE NODE MODEL APPROACH

1. Concept Behind Modification From Six Node Model

The computer simulations conducted on the six node model encountered the same problem in each simulation. The values of the unknowns a_{56} and a_{65} , corresponding to the mount location, would jump a large amount in value causing the program differential equation solver to experience a fatal error, indicating that the equations were too stiff. Changing the relationship between the convective rate coefficients, b_{ij} , had no effect on the resultant outcome and the jump in these unknowns prevented the convergence of the program. These unknowns contribute to the temperature profiles primarily at the mount location. The convective heat transfer coefficients that were to be identified were between the free stream and nodes two, three and four of the vane and at this point we were not concerned with the mount. Examination of the experimental time temperature profiles at the mount location, for the full scale vane, revealed that the temperature at these locations had little variation. The temperature varied less than 25°F over the entire 0.7 second rocket firing period as compared to the temperature variation at nodes two, three, and four which varied from 900 to 3000° F during the same rocket firing time. In contrast to the quarter scale modeling the temperature variation in the mount, node four, was significant. The size of the quarter scale vane limited the

locations for attachment of thermocouples to the shaft and the mount. This limited the experimental temperature data to only two locations, the mount and shaft, making it necessary to use the mount temperature profiles in the matching process. The full scale vane is larger in size and provides greater space to place thermocouples and collect the experimental data.

The model was constructed to determine the convective rate coefficients from the free stream to the vane. Due to the small variation of temperature in the mount and the enormous change of the unknowns using the mount node it was decided that a modification needed to be made to the six node model. This modification was the elimination of the mount node, which eliminated the two unknowns, a_{56} and a_{65} that were causing the program to fail. There was also another motivation for eliminating the mount node, which was the inconsistency in the use of the mount apparatus for all the experimental rocket firings [Ref. 1:p. 13]. The modification to a five node model is illustrated in Figure 3.8.

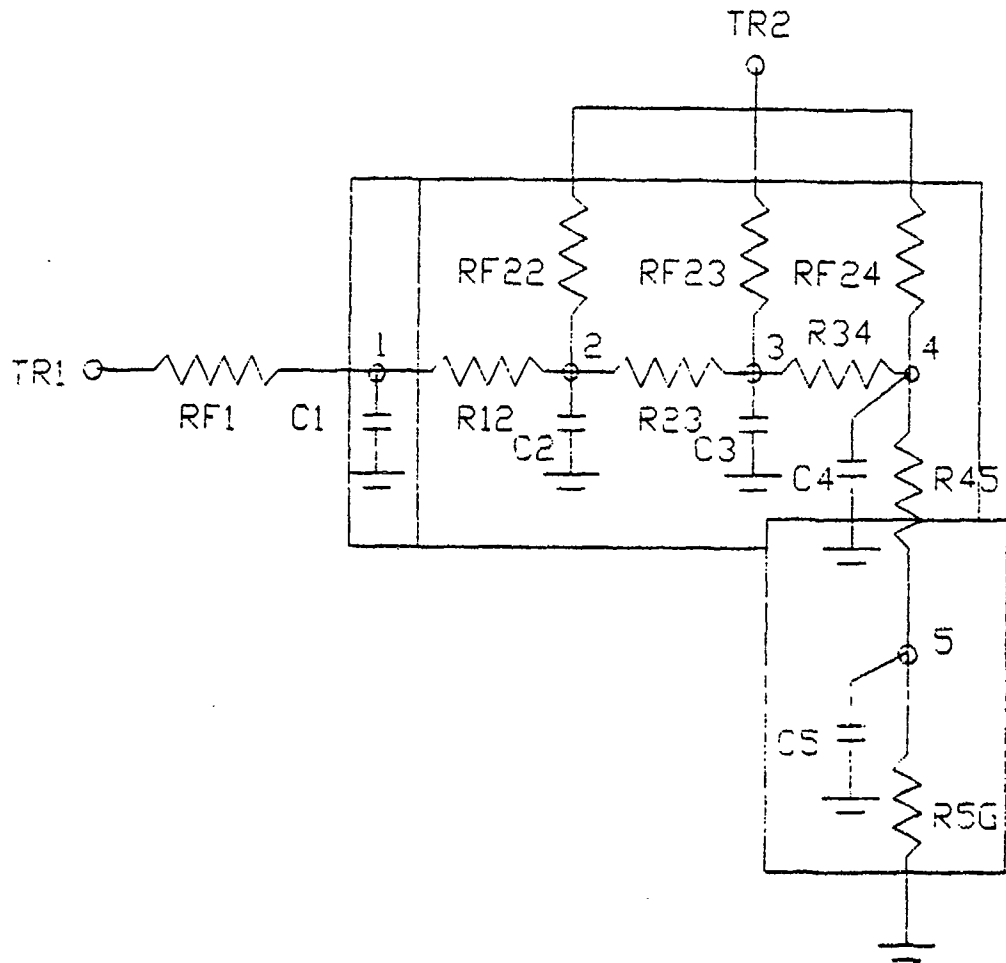


Figure 3.8 Five Node Vane Model

2. Energy Balance Equation Formulation

The formulation of the energy balance equations for the five node model was done using the same process as that for the six node with only minor changes. The modification involves eliminating node six which in turn eliminates the last equation from the six node model energy balance. There was also only a minor change to the equation for node five of

the original six node equations. The following equations are applicable to the five node vane model:

$$\dot{T}_1 = -a_{11}T_1 + a_{12}T_2 + b_{11}T_{R1} \quad (45)$$

$$\dot{T}_2 = a_{21}T_1 - a_{22}T_2 + a_{23}T_3 + b_{22}T_{R2} \quad (46)$$

$$\dot{T}_3 = a_{32}T_2 - a_{33}T_3 + a_{34}T_4 + b_{32}T_{R2} \quad (47)$$

$$\dot{T}_4 = a_{43}T_3 - a_{44}T_4 + a_{45}T_5 + b_{42}T_{R2} \quad (48)$$

$$\dot{T}_5 = a_{54}T_4 - a_{55}T_5 \quad (49)$$

$$a_{5G} = \frac{1}{C_5 R_{5G}} \quad a_{55} = a_{54} + a_{5G} \quad (50)$$

Reduction of the model to five nodes eliminates the need to identify the unknown conductive resistances from the shaft to the mount, R_{56} , the mount to ground, R_{6G} , and the capacitance of the mount node, C_6 . The experimental data for the five node full scale modeling process will be from the thermocouple attachment points that correspond to nodes two, three and four of the vane model.

3. Program Modification

The main program PID needs to be modified from the six node model to that of the five node. This is done primarily by making an alteration in the FCN subroutine which generates the differential equations to be used. The only other modification that needs to be made to the PID program is the set up of the size of the dimensions for the rate coefficients

and the setting of the initial conditions. Identification of the unknowns will be discussed in the following section. The FORTRAN coding used to generate the simulation program was constructed to be flexible in that it allows for changes in the vane nodes without major program modification.

4. Model Simulation ZXSSQ Optimizer

a. Unknowns and Assumptions

The modeling process for the five node model will be set up in the same way the six node model utilized the ZXSSQ optimizer. Prior to starting the simulations the unknowns for this model need to be identified. As before the unknowns are the convective rate coefficients, b_{22} , b_{32} and b_{42} , along with the conductive rate coefficient, a_{5g} . In the quarter scale modeling it was demonstrated that the convective rate coefficient from the tip, b_{11} , had little effect on the determination of the unknown parameters. Without knowing the effect on the full scale model the value of b_{11} will be considered a constant for the first series of simulations. The assumption of dependence between the convective rate coefficients was applied to the first series of simulations making $b_{22}=b_{32}=b_{42}$. A second set of simulations will be conducted making the values of the convective rate coefficients independent of each other and making the tip convective rate coefficient a variable.

b. Experimental Data Used

The experimental data used for the five node vane model will be from the thermocouple locations that correspond to nodes two, three, and four of the model. For the simulations involving the use of the ZXSSQ optimizer, an attempt will be made to match all three time-temperature profiles. A point to be noted is that the values of the recovery temperatures that are to be utilized are different from those used in the quarter scale model due to the different rocket motor used in the test firings. The temperatures used to drive the heat transfer process were $T_{R1}=2970$ and $T_{R2}=2870$ Kelvin, respectively [Ref. 2:p. 44]. Referencing these temperatures to the ambient, as was done with the experimental node temperatures, would lead to recovery temperatures being normalized to values of $T_{R1}=2670$ and $T_{R2}=2570$ Kelvin.

c. Results

As seen with the six node modelling process, using the ZXSSQ optimizer, the model failed to converge. Changing the assumption for the dependence of the convective rate coefficients, which made the values of the convective rate coefficients independent variables had no effect, and the program still failed to converge. An attempt was made to match a single node time temperature profile but this also proved fruitless and again the program failed to converge.

From the foregoing results it is evident that the use of the ZXSSQ optimizer for the full scale modeling process is inadequate. There is no provision in the ZXSSQ optimizer to control the values or range of the unknown variables.

5. Model Simulation DBCLSF Optimizer

a. Constrained Model Simulation

As with the six node model simulations, a constrained optimizer replaced the present optimizer, ZXSSQ. This was an attempt to hold the unknowns within certain limits permitting the optimizer a chance to avoid or recover from any possible local minimum and prevent the unknowns from becoming so large that the program fails to converge. The DBCLSF constrained optimizer utilized will be from the IMSL library. The first simulation assumed that the convective rate coefficients b_{22} , b_{32} and b_{42} are equal and that b_{11} was a constant and attempted to match the temperature profiles of nodes two, three, and four. This simulation also failed to converge. The assumption that b_{11} was a constant was changed and b_{11} was allowed to vary. This simulation also failed to converge.

b. Single Node Temperature Profile Modeling

A different approach was taken to try and establish a five node working model of the vane. Instead of trying to match all three temperature profiles the model was altered to try and match only a single node time-temperature history then

work up to two nodes and then a full three node matching process. The assumptions, for the single node temperature matching process, that any of the unknown variables were dependent were abandoned and each unknown variable was considered independent of the others.

(1) *Node Two Temperature Modeling.* The modeling simulations began by trying to match the time-temperature history of only a single node, node two. This was attempted in the hope that if only one profile could be matched that by examination of the profiles generated for the other nodes a slight adjustment could be made in order to obtain a match between all three node time temperature profiles. The program was modified to match the profile of node two and the boundary limits were set to 0.0001 for the lower boundary and 100 for the upper boundary, with recovery temperatures of $T_{R1}=2670$ and $T_{R2}=2570$, and the initial conditions set to zero. The simulation converged with a sum of the squares error of 12.01, with values for the unknowns $b_{11}=3.0833$, $b_{22}=0.0255$, $b_{33}=0.0492$, and $b_{44}=0.2496$. The value for a_{10} reached the lower limit of 0.0001 set in the constrained optimizer and did not vary from that point. Figure 3.9 illustrates the time-temperature profiles for this simulation.

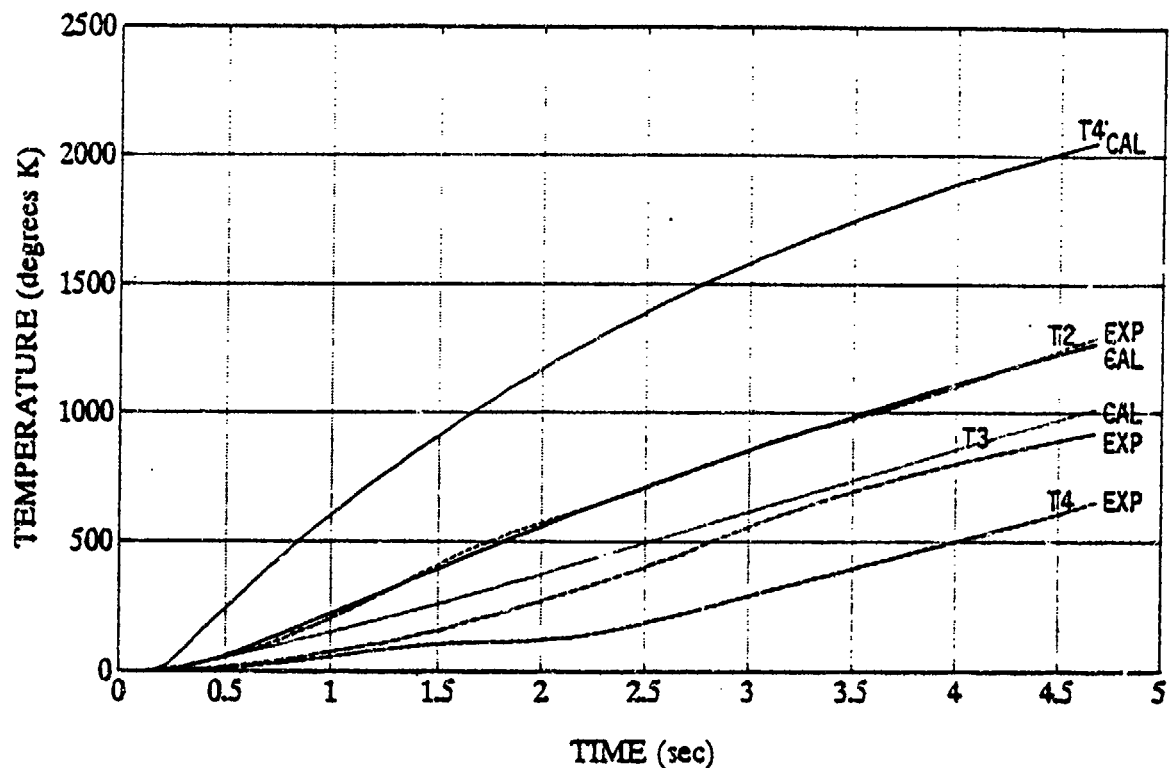


Figure 3.9 Time-Temperature Profiles Node Two Match

(2) Node Three Temperature Modeling. The main program PID was modified to match experimental data for node three only. The program simulation converged with a sum of the squares error of 15.26. The values obtained for the variables were $b_{11}=2.7883$, $b_{22}=0.8934$, $b_{32}=0.0001$ and $b_{42}=0.1778$. Again the value for a_{50} dropped to the lower limit allowed by the constrained optimizer, 0.0001, and remained there. Figure 3.10 illustrates the time-temperature profiles of the node three match.

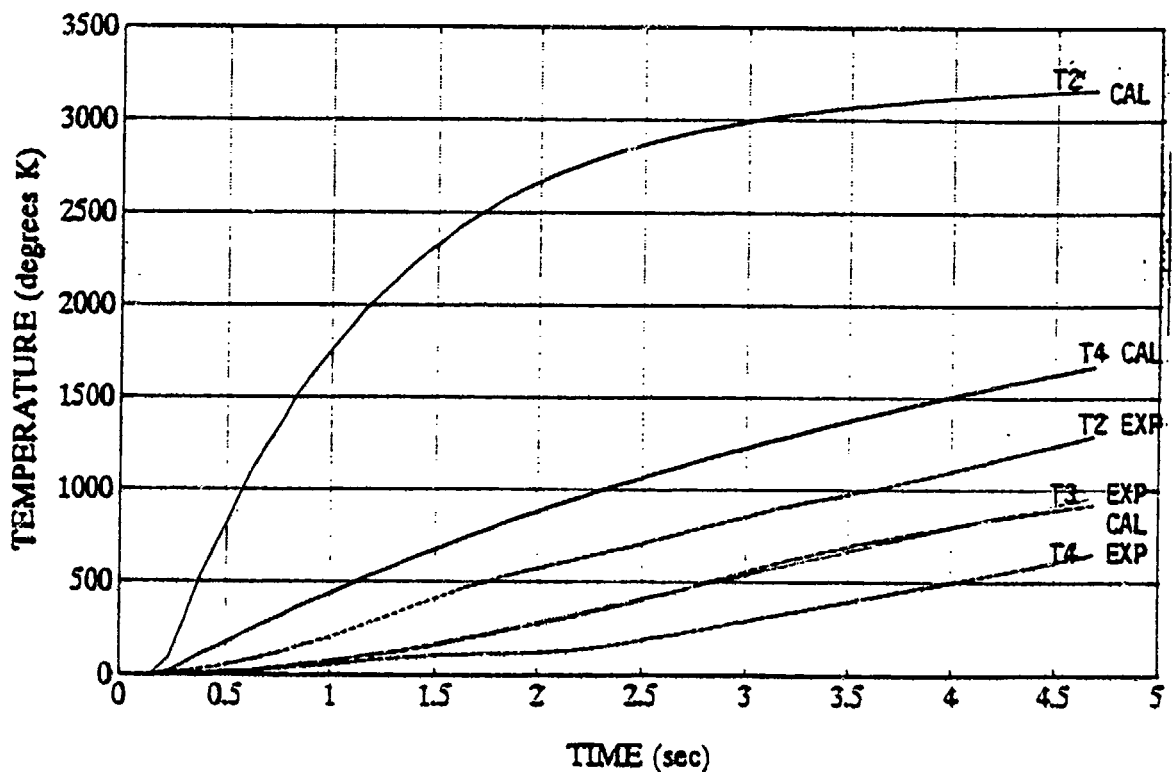


Figure 3.10 Time-Temperature Profiles Node Three Match

(3) *Node Four Temperature Modeling.* Again the program was modified to match the experimental data for node four only and the simulation converged for this set up with a sum of the squares error of 39.539. The values obtained for the unknowns were $b_{11}=2.92$, $b_{22}=1.38$, $b_{33}=0.212$ and $b_{44}=0.02$ and again the value of a_{50} went to the lower boundary limit set in the constrained optimizer and remained there. The time-temperature profiles are illustrated in Figure 3.11.

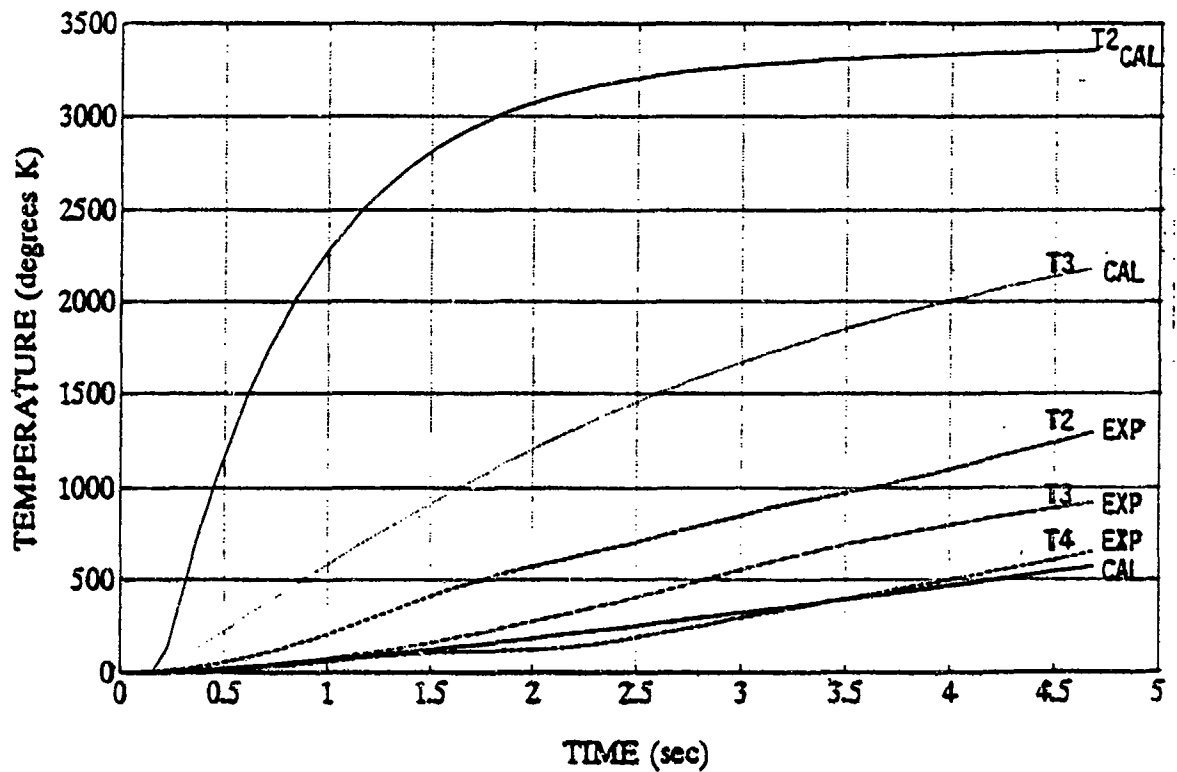


Figure 3.11 Time-Temperature Profiles Node Four Match

(4) Results. In all three simulations, using only a single set of experimental data, the program converged and the calculated and experimental profiles of the node that was being matched were as accurate as the optimizer could get. The error using node two data was less than that of node three and much less than that of node four. The problem arises in the calculated temperature histories of the other two nodes not being matched. It can be seen in Figures 3.9, 3.10 and 3.11 that the two nodes not being matched are significantly different from the experimental data. In the simulation using the experimental data from node two the profiles of node two

and three match fairly closely while the profile for node four is nowhere near the experimental data. In the simulation using the node three experimental data, node three profiles have a fair match but the profiles for nodes two and four are unsatisfactory. The same condition holds when using the experimental data for node four only. It must be noted that the magnitude of the error between the calculated and experimental data is dependent on the quality of the experimental data. If the data follows a first order linear response closely then the match of the calculated and experimental profiles is adequate and the sum of the squares error is low. If the experimental profiles vary from an exponential curve then the first order linear model matches the experimental profile the best it can and the error will be large.

c. Manual Adjustment of Variables-Forward Model

The results of the single node temperature profile simulation permit the use of the forward model to generate a series of time-temperature profiles with the final values obtained for the unknowns. By adjusting the various unknowns we can see the effect one variable has on another. A set of values needs to be selected for the unknowns. These values can be arbitrary since we are looking at the effects of one value on another. The initial starting values for the unknowns were chosen as, $b_{11}=1.76$, $b_{22}=0.0455$, $b_{32}=0.06$, $b_{42}=0.02$

and $a_{5g}=0.0001$. The value of a_{5g} was chosen as 0.0001 because in all the previous simulations this unknown jumped to this lower value limit and remained there. The values for the other unknowns were chosen by scaling up the final values of the quarter scale model results obtained by Reno [Ref. 1] and considering the values obtained from the single node temperature profile results. The effects of the variables were that, the value of b_{11} influenced the temperature profile of node two and had little or no effect on nodes three and four, while the same holds true for the other convective rate coefficients. Altering each one consecutively varied the result of that particular node profile but had little effect on the outcome of the other node calculated temperature profiles. As an example of this effect the values of the unknowns listed above were used in the forward model. The model used a set of recovery temperatures with values of $T_{R1}=2970$ and $T_{R2}=2870$. There was no need in normalizing these recovery temperatures in the forward model. The effects of one variable on the other was being examined and not actually trying to determine the proper values for the unknowns as done in the PID modeling program. The profiles generated by this forward model simulation are illustrated in Figure 3.12 where the three nodes of concern, nodes two, three, and four are displayed.

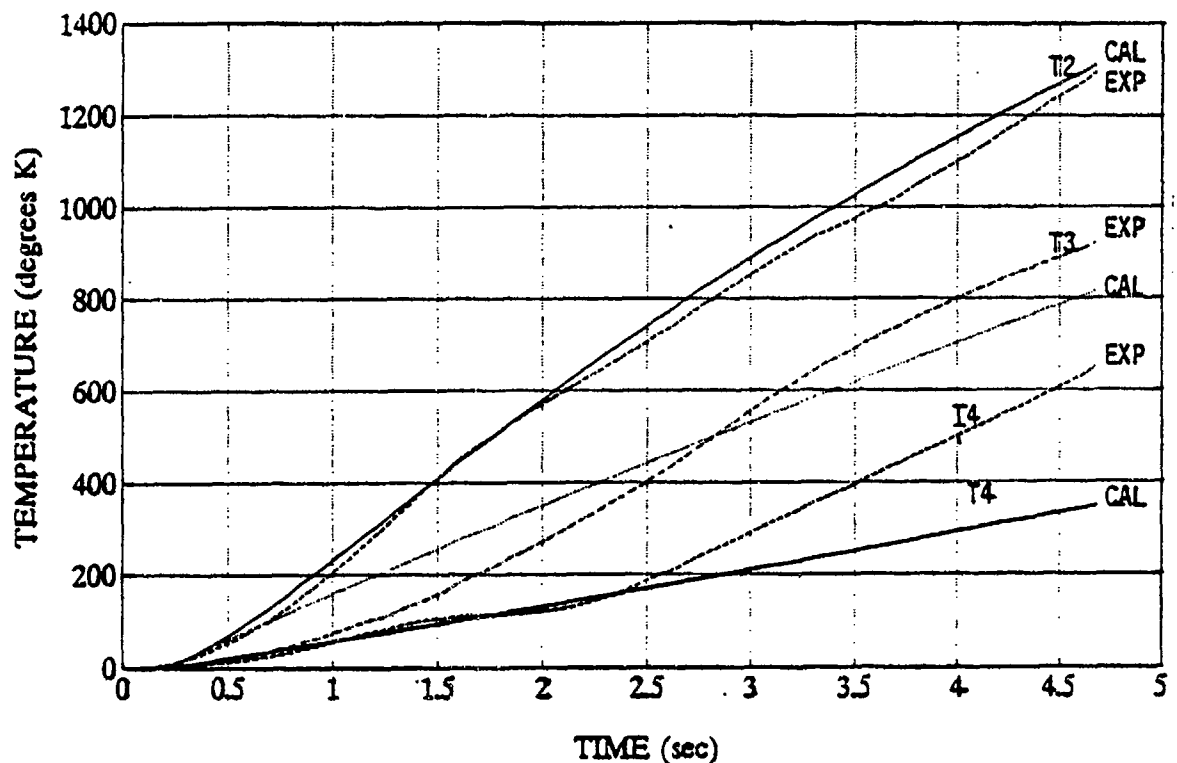


Figure 3.12 Forward Model Time-Temperature Profiles

These profiles show a good match between the calculated and experimental temperature profiles of node two. The profiles of nodes three and four do not follow the experimental data but again the experimental data for these two nodes does not follow an exponential curve. Another thing that needs to be pointed out is the calculated curves for the three nodes appear to be linear straight lines. This is not an exponential curve as expected from a first order linear model. To show the effects of changing one variable on the other temperature profiles, the value of b_2 was changed to 0.03 and the forward model simulation was executed. The resulting time-temperature profiles are displayed in Figure 3.13.

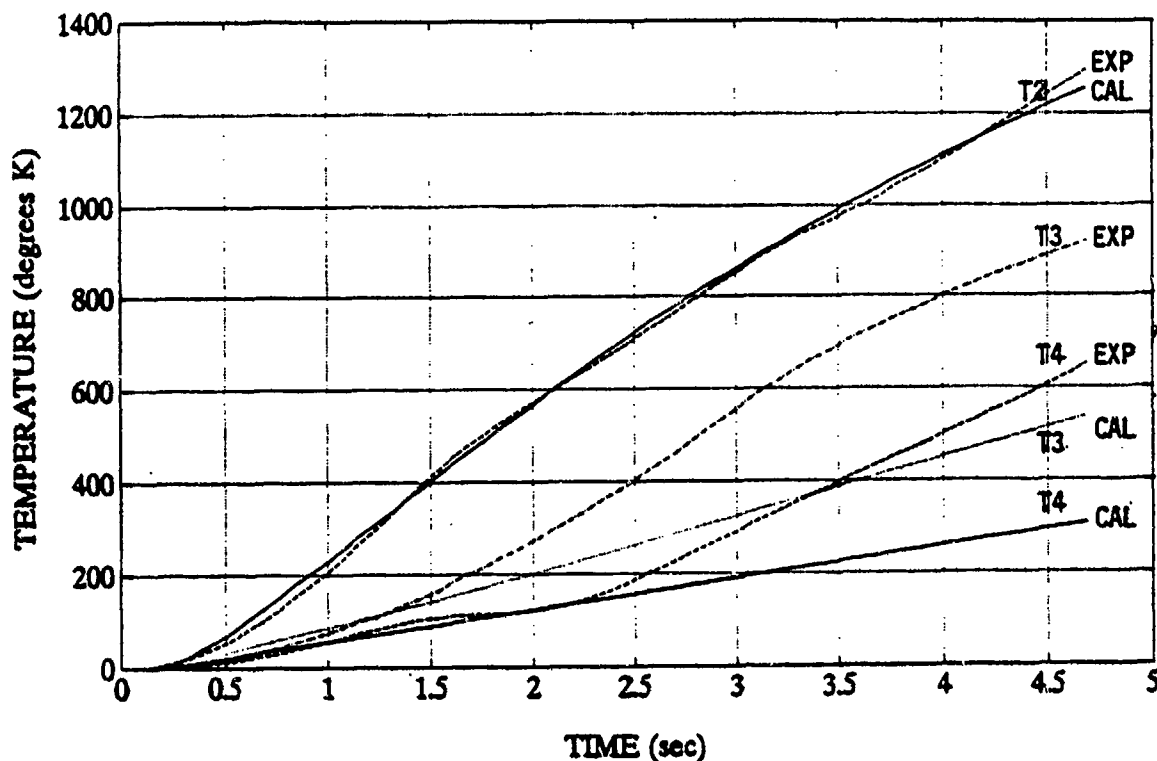


Figure 3.13 Forward Model Simulation Node 3 Variant

Comparison of Figures 3.12 and 3.13 show that there was a significant change in the calculated temperature profile of node three, while the profiles of nodes two and four changed very little in magnitude compared to node three. The program calculated the sum of the squares error for both simulations. The error for the first simulation was 74.4 while the error for the second simulation was 213.577.

Adjustments can be made to each of the unknown variables in the forward model to create a better match for any of the individual curves. This process of adjusting the unknowns is a manual attempt to do what the optimizer was intended to do in the PID program. The manual adjustment

allows the user to view the effects that individual rate coefficients have on the temperature profiles of the other nodes.

d. Two Node Temperature Profile Modeling

The previous section established the convergence of the modeling program and the matching of a single time-temperature profile. The model will be upgraded to matching two temperature profiles in an attempt to obtain a three node full scale matching model.

(1) *Nodes Two and Three Temperature Modeling.* The modeling program was modified to match the temperature profiles of two nodes, vice one, with the sum of the squares error being calculated based on both temperature profiles. The first simulation will attempt to match the temperature profiles of nodes two and three. The assumptions for the two node matching simulations are the same as those for the single node in that all the unknowns are considered independent of each other and the constrained optimizer will be utilized with the upper boundary of 100 and a lower boundary of 0.0001. The simulation converged with the resulting sum of the squares error of 15.217 and values for the unknowns $b_{11}=2.83$, $b_{22}=0.0297$, $b_{32}=0.0056$, $b_{42}=0.9408$ and $a_{5g}=0.0001$. Again the value for a_{5g} jumped to the lower limit and remained there while the optimizer was changing the values for the other

unknowns. The time-temperature profiles for the two node matching process are illustrated in Figure 3.14.

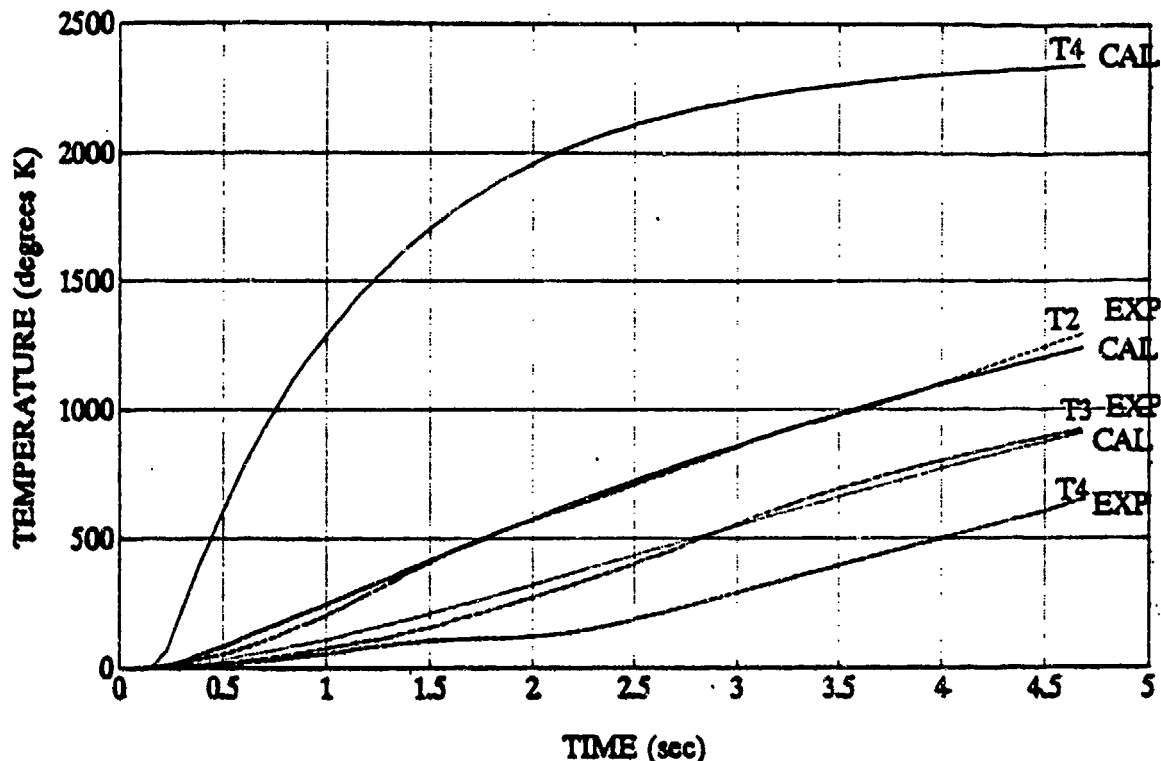


Figure 3.14 Two Node Temperature Profiles

The simulation program matched the time-temperature profiles for nodes two and three while the calculated profile for node four was no where near the experimental profile. Figure 3.14 illustrates that the calculated profile for node four, T4 calculated profile, was unsatisfactory.

(2) Nodes Two and Four Temperature Modeling. The program was again modified to match the data for the temperature profiles from nodes two and four. The simulation using these experimental profiles yielded a sum of the squares

error of 216.02 and values of the unknowns $b_{11}=1.2010$, $b_{22}=0.1151$, $b_{32}=0.0001$, $b_{42}=0.0001$ and $a_{5G}=0.0001$. The time temperature profiles for this simulation are illustrated in Figure 3.15.

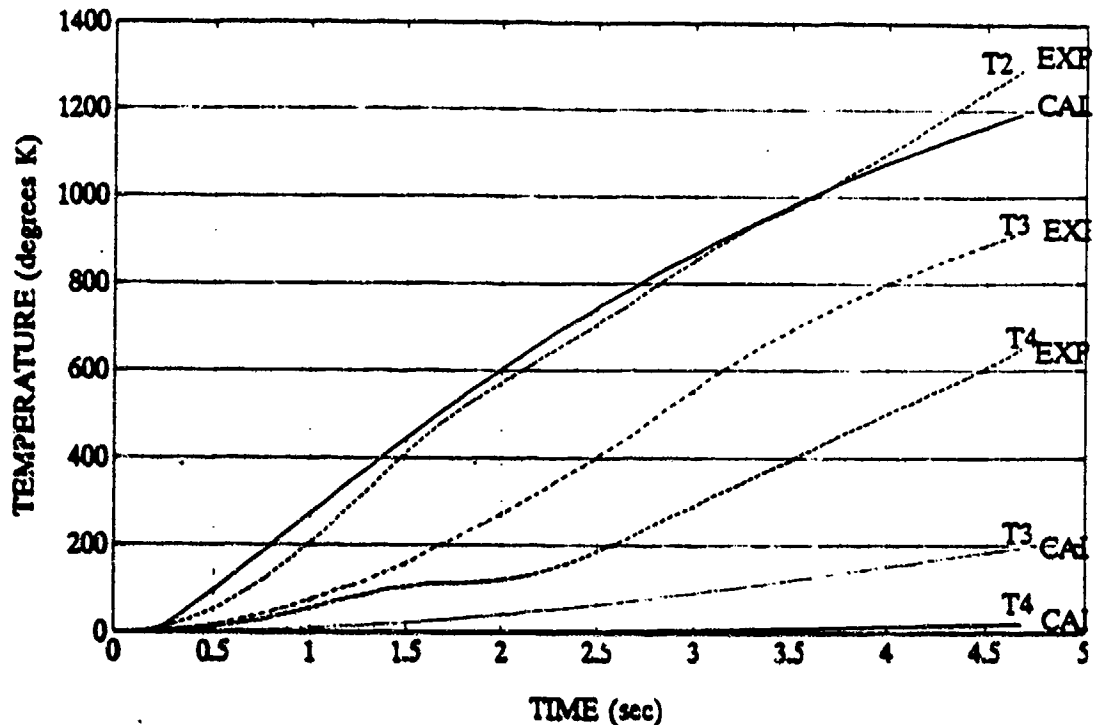


Figure 3.15 Two Node Temperature Profiles

The experimental and calculated profiles for node two matched adequately, but the calculated profile for node four is substantially off from the experimental data. The value of a_{5G} again reached the lower limit set by the optimizer and remained there.

(3) *Nodes Three and Four Temperature Modeling.* The final two node matching simulation was conducted by modifying the main program and calculating the sum of the squares error

based on the temperature profiles from nodes three and four. The simulation yielded values of $b_{11}=2.19$, $b_{22}=0.77$, $b_{32}=0.06$, $b_{42}=0.0001$ and $a_{5G}=0.0001$ with a sum of the squares error of 185.95. As can be seen in Figure 3.16 the program adequately matched the profiles for nodes three and four but there was a significant difference in the calculated and experimental time temperature profile of node two.

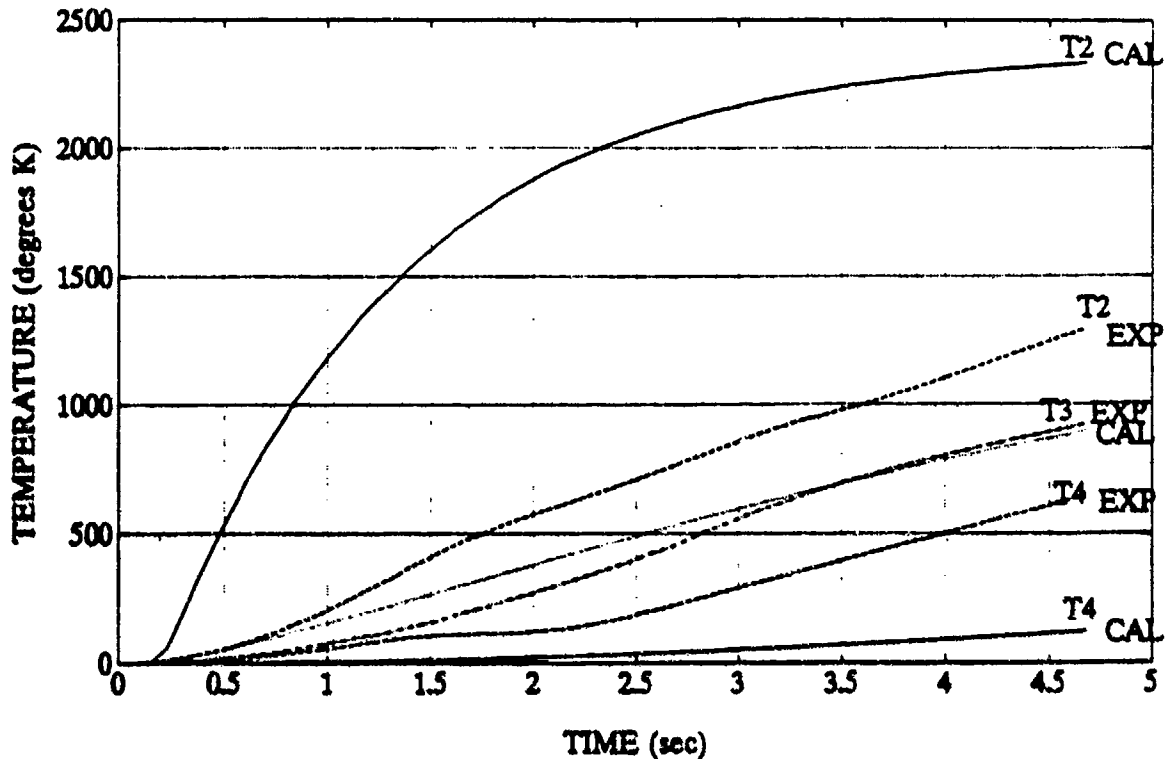


Figure 3.16 Two Node Temperature Profiles

(4) *Results.* The program was able to match the temperature profiles from node two and to a lesser extent node three, but had difficulty in matching the profiles for node four. This is due to the fact that the experimental profile for node four is not an exponential curve. The first order

linear modeling program provides an descent attempt to model the experimental data and provides as close an estimation possible. The use of the node four experimental data causes a significant increase in the sum of the squares error. Although it may be possible to get an adequate match for two temperature profiles the third is by far not satisfactory.

e. Convergent Simulation Model

(1) *Three Node Temperature Profile Matching.* It has been demonstrated that the modeling program can match one and two temperature profiles. The process is now upgraded to model all three temperature profiles of the vane, that is the modeling of nodes two, three, and four. This is the primary reason behind the modeling of the full scale vane.

(2) *Unknowns and Assumptions.* The model simulation for the three node temperature matching will assume that the unknown variables, b_{11} , b_{22} , b_{32} , b_{42} and a_{5G} , are independent of each other. Previous five node model simulations considered b_{11} a constant in the same manner as the quarter scale modeling did. Changing the value of b_{11} in the quarter scale model had little effect on the values of the other unknowns. This fact has not been established in the full scale modeling process which was the reasoning behind making b_{11} a variable. The initial conditions will be set to zero for all five unknowns and the values for the recovery temperatures are $T_{R1}=2670$ and $T_{R2}=2570$ Kelvin. For this

simulation the sum of the squares error were be calculated based on all three temperature profiles and the boundary limits for the constrained optimizer were set to 0.0001 for the lower boundary and 100 for the upper boundary.

(3) *Results.* The simulation was executed for a time period of zero to five seconds. The simulation converged with a sum of the squares error of 58.5 and the values obtained for the unknown variables were $b_{11}=1.3787$, $b_{22}=0.0938$, $b_{32}=0.0862$, $b_{42}=0.0484$ and $a_{5G}=0.0001$. As with the single and two node temperature matching the value for a_{5G} dropped to the lower boundary set in the optimizer. The low value obtained for a_{5G} relates to the negative value obtained for the similar unknown a_{4G} of the quarter scale model from Driels [Ref. 3] simulation. The constrained optimizer prevented the value for a_{5G} from going negative in the five node model simulation. The time-temperature profiles generated by this simulation are illustrated in Figure 3.17.

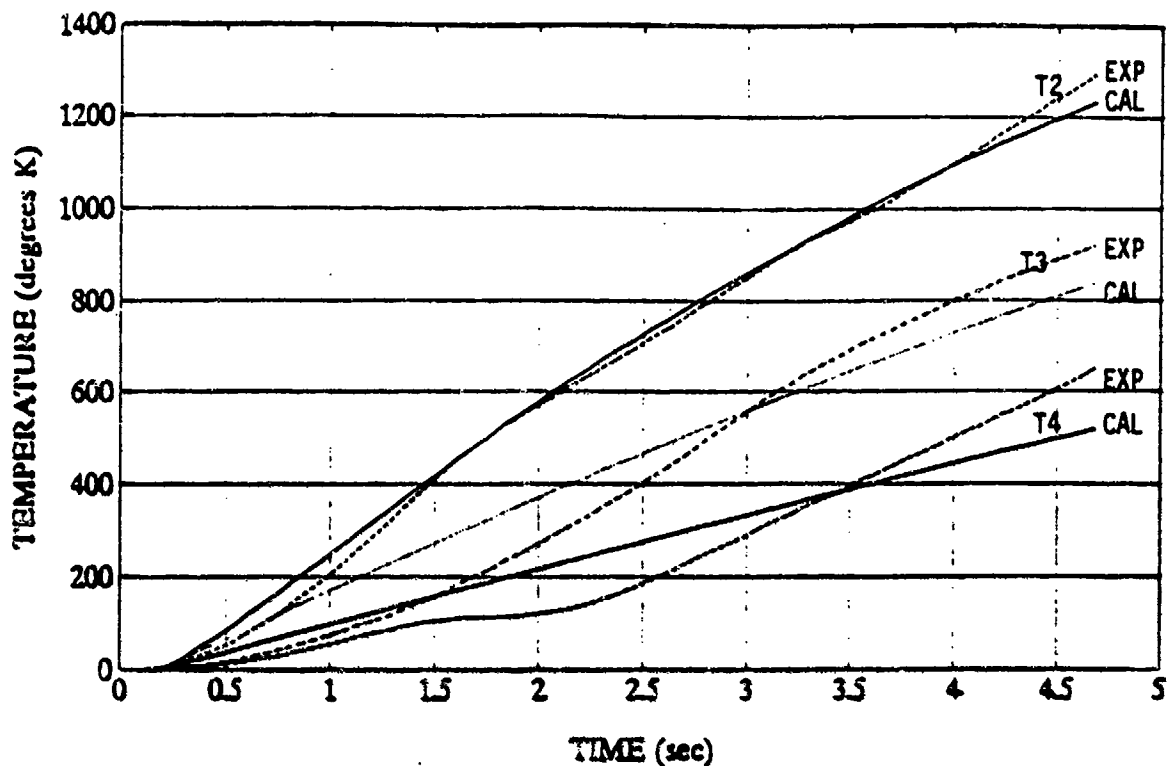


Figure 3.17 Five Node Vane Model Temperature Histories

The match between the profile labeled T2, for node two, was an adequate match as it was in some of the previous simulations. The match between the experimental and calculated profiles for nodes three and four were not as close as node two. Again this is due to the quality of the experimental data taken. If the data does not follow an exponential rise such as a first order linear system, then the optimizer is less accurate in matching the experimental temperature profiles.

The simulation above was conducted using the recovery temperatures of $T_{R1}=2670$ and $T_{R2}=2570$ Kelvin. These values were obtained from past work conducted by Nunn [Ref. 1] and Reno [Ref. 2]. Since there was no direct way to determine these values for the full scale experimental rocket motor

firings, due to the unavailability of firing data, the values of the recovery temperatures were altered to illustrate the effects on the model simulation. The first modification was to change the values of the recovery temperatures to the value of the chamber temperature. This would make $T_{R1}=2970$ and $T_{R2}=2870$ degrees Kelvin. This simulation converged with a sum of the squares error of 56.5. The values of the unknowns were $b_{11}=1.4747$, $b_{22}=0.0716$, $b_{32}=0.0761$, $b_{42}=0.0430$ and $a_{5G}=0.0001$. The time-temperature profiles were very similar to those in Figure 3.17. Another simulation was conducted with the values of $T_{R1}=3550$ and $T_{R2}=3450$ degrees Kelvin. This simulation yielded an error of 54.5 and values of $b_{11}=2.3045$, $b_{22}=0.033$, $b_{32}=0.0621$, $b_{42}=0.0354$ and $a_{5G}=0.0001$. The time-temperature profiles for this simulation are again very similar to those obtained in Figure 3.17. This would be expected due to such a small variation in the sum of the squares error.

f. Simulation With Various Initial Conditions

To test the robustness of the simulation program the initial conditions were varied. Changing the initial conditions changes the starting point for the model simulation. Variation of the starting values for the unknowns yielded the same final values. The program converged to the same point and the same profiles when the initial conditions were changed to different values.

g. Calculations of Heat Transfer Coefficients

The values obtained for the unknown variables, for the three node temperature matching simulation with the recovery temperatures of $T_{R1}=2670$ and $T_{R2}=2570$ Kelvin, were $b_{11}=1.3787$, $b_{22}=0.0938$, $b_{32}=0.0862$, $b_{42}=0.0484$ and $a_{5G}=0.0001$. The convective heat transfer coefficients can now be determined but the values for the resistance must first be determined. These calculations yield $R_{F1}=0.1044$ K/W, $R_{F22}=0.3101$ K/W, $R_{F23}=0.2530$ K/W and $R_{F24}=0.3605$ K/W with the values for the corresponding convective heat transfer coefficients; $h_t=22027.5$ W/m²•K, $h_{v2}=1150$ W/m²•K, $h_{v3}=1057$ W/m²•K, and $h_{v4}=594$ W/m²•K. Where h_t is the convective heat transfer coefficient for the tip and h_{v2} , h_{v3} and h_{v4} are the convective heat transfer coefficients from the free stream to nodes two, three and four of the vane, respectively.

The simulation involving the use of $T_{R1}=2970$ and $T_{R2}=2870$ Kelvin yielded the values of $b_{11}=1.4747$, $b_{22}=0.0716$, $b_{32}=0.0761$, $b_{42}=0.0430$ and $a_{5G}=0.0001$. These values yield resistances of $R_{F1}=0.0976$ K/W, $R_{F22}=0.4062$ K/W, $R_{F23}=0.287$ K/W, and $R_{F24}=0.4058$ K/W with the resulting convective heat transfer coefficients of $h_t=23561.3$ W/m²•K, $h_{v2}=877.8$ W/m²•K, $h_{v3}=932.86$ W/m²•K and $h_{v4}=527.74$ W/m²•K. The last simulation involved changing the recovery temperatures to $T_{R1}=3550$ and $T_{R2}=3450$ Kelvin. The results of this simulation yielded $b_{11}=2.1045$, $b_{22}=0.033$, $b_{32}=0.0621$, $b_{42}=0.0354$ and $a_{5G}=0.0001$. The resistance values obtained from these results are

$R_{F1}=0.0624$ K/W, $R_{F22}=0.8815$ K/W, $R_{F23}=0.3512$ K/W and $R_{F24}=0.4928$ K/W. Using these resistances the values for the convective heat transfer coefficients are $h_t=36819$ W/m²•K, $h_{v2}=404.6$ W/m²•K, $h_{v3}=761.5$ W/m²•K and $h_{v4}=434.5$ W/m²•K.

DISCUSSION OF RESULTS

The full scale six node model attempted to directly scale up the four node quarter scale vane model. The decision to add the two extra nodes was necessary in order to model the larger vane and was believed to more accurately describe the heat transfer process for the larger amount of material contained in the full scale vane. The first series of simulations involved the use of the ZXSSQ optimizer to adjust the unknowns in order to obtain a match between the experimental and calculated time-temperature profiles. The unknown rate coefficients for the six node model were identified as b_{22} , b_{32} , b_{42} , a_{56} , a_{65} and a_{60} . The assumptions were that b_{11} was a constant, as it was in the quarter scale simulations, and that the convective rate coefficients, b_{22} , b_{32} and b_{42} , were equal. These series of simulations failed to converge due to the substantial variation in the unknown values, a_{56} , a_{65} and a_{60} which contain the resistances between the shaft and mount and the mount and ground. The values of a_{65} and a_{60} also contain the unknown value of the mount node capacitance. During the simulation these variables jumped substantially in value causing the differential equations to become stiff. The jump in the variables were both in the negative and positive direction. The experimental data for these simulations was taken from the shaft and mount locations

in the same manner that the experimental profiles were used in the quarter scale modeling process. When the simulation failed to converge the experimental data was changed to that relating to nodes two, three and four of the six node model. In the physical test firings, for the full scale vane, the temperature profiles corresponding to the shaft and mount locations had minimal variation from the ambient. It was believed that the simulation program had difficulty adjusting to the small changes in the mount and shaft temperature profiles while trying to make large adjustments to the unknowns from nodes two, three and four. That is, over a time increment the temperature change was small in the shaft and mount but was large in the tip and fin area. This idea led to abandoning the use of the experimental data from the shaft and mount and using the experimental data from the fin area nodes, nodes two, three and four. The simulation using this new set of experimental profiles also failed to converge with the same results for the unknowns, a_{56} , a_{65} and a_{60} .

Another idea considered controlling the simulation by preventing the unknown values from varying substantially. This was achieved by the use of the constrained optimizer, which would permit the setting of boundary limits to hold the unknowns within a certain range. This range was selected as positive values with a lower boundary limit of 0.0001 and an upper boundary limit of 100. The assumptions for the unknowns remained the same as the previous simulations. This

simulation using the constrained optimizer also failed to converge, where the values of the unknowns a_{56} , a_{65} and a_{66} reached the set limits and remained there. Attempts were made at raising the upper boundary in the hope that the higher boundary limit would allow the optimizer a chance to avoid or recover from any local minimum and continue on with the simulation. Raising the upper boundary limit had the same result as before, the unknown values reached the boundary limits and remained there and the program failed to converge. Any attempt at setting the lower boundary limit to a negative value or raising the upper limit to too high a value caused an arithmetic fault to occur. A set of simulations was conducted using each optimizer but changing the assumptions, making the convective rate coefficients b_{22} , b_{32} , and b_{42} independent of each other. In both cases these simulations failed to converge.

Failure to resolve the problem of the unknowns associated with the shaft and mount led to the modification of the model from six nodes to five. This was done by eliminating the mount node which eliminated the unknowns a_{56} , a_{65} and a_{66} and added the unknown a_{50} . This was justified by the fact that there was no consistent mount arrangement used from one test firing to another. The unknown heat transfer coefficients that are being determined are from the free stream to the fin area of the vane, not the mount or shaft. The small temperature change in the experimental profiles indicated that

very little energy from the fin section was being transmitted to the shaft and mount. The unknowns were identified as b_{11} , b_{22} , b_{32} , b_{42} and a_{50} . The assumptions, for the first series of simulations, were that b_{11} was a constant and $b_{22}=b_{32}=b_{42}$. With the program modification in place the simulation was conducted using the ZXSSQ optimizer. The result of this simulation was the same as in the six node model, the program failed to converge using this optimizer. Attempts were made at using only a single experimental temperature profile, but these simulations also failed to converge. The five node program was modified as was the six node to use a constrained optimizer to place a set of boundary limits on the unknowns with the limits set to the same values as in the six node simulations. The assumptions for the five node model unknowns remained the same with b_{11} constant and $b_{22}=b_{32}=b_{42}$. This simulation yielded the same results as the previous modeling attempts in that it failed to converge. A more systematic approach was taken to determine if the simulations could be made to converge. To accomplish this the simulations would be constructed to match only one temperature profile at a time. The assumptions were that none of the unknowns were dependent on one another and the term b_{11} was no longer considered a constant as before. The unknown variables were identified as b_{11} , b_{22} , b_{32} , b_{42} and a_{50} .

The simulations converged and succeeded in providing an adequate match for the selected single node time temperature

profiles. The other two fin area node profiles were unsatisfactory in the match between the experimental and calculated profiles. The model was modified to match two temperature profiles from the fin area of the vane. This series of simulations converged and an adequate match was achieved for two of the three temperature profiles, but again the problem was in the third profile. The program matched the two temperature profiles that it was set up to match but the calculated profile for the third node was substantially off from the experimental data. Being able to match two node temperature profiles the program was modified in an attempt to match all three nodal temperature profiles. The program was modified again to read experimental data for nodes two, three and four, and to match all three time temperature profiles. This simulation was conducted using the assumptions that b_{11} , b_{22} , b_{32} , b_{42} , and a_{50} were independent, the recovery temperatures were $T_{r1}=2670$ and $T_{r2}=2570$ Kelvin, and the initial conditions set to zero. This simulation converged and yielded the profiles displayed in Figure 3.17. The simulation was tested for its robustness by starting with various initial conditions. The program converged to the same set of values with the same set of time-temperature profiles. The match between the calculated and experimental profiles of node two were satisfactory while the match for nodes three and four could only be considered adequate. The deficiency here is not with the modeling program but with the experimental time-

temperature profiles. The experimental data does not follow an exponential curve, where as the modeling program is a first order linear model and can only match exactly an exponential profile. The modeling program matches these profiles the closest it can to an exponential curve. Another point to note is the calculated temperature profiles for all three nodes are almost a straight line with very little curve in the calculated values in any of these three temperature profiles. Since the exact recovery temperatures were unknown the three node temperature matching simulation was conducted using two other sets of profiles. Table 4 lists the recovery temperatures used and the corresponding convective heat transfer coefficients.

TABLE 4

Recovery Temperatures T_{R1}/T_{R2}	Convective Heat Transfer Coefficient $W/m^2 \cdot K$			
	h_c	h_{v2}	h_{v3}	h_{v4}
2670/2570	22027	1150	1057	594
2970/2870	23562	878	933	527
3550/3450	36819	405	762	434

The values of the heat transfer coefficients displayed in Table 4 are of the same magnitude as those obtained by Driels in the quarter scale simulations. The values for the heat transfer coefficients from quarter scale modeling were $h_c=30122 W/m^2 \cdot K$ and $h_v=210 W/m^2 \cdot K$ [Ref. 3:p. 12].

The calculated profiles in the convergent five node model simulation were nearly straight lines. This effect was caused by the reduction of the conductive rate coefficients, the a_{ij} values. These values are determined from equations (2) and (4), the resistance and capacitance equations. Consider the area A as a length squared and the volume V as a length cubed. The value of a_{ij} is equal to $(RC)^{-1}$, and inserting the equivalent values from equations (2) and (4), and considering the physical properties as constants, the value of RC becomes a function of the length squared. This means that the value of a_{ij} is a function of the $(\text{length}^2)^{-1}$. As the quarter scale model is increased in size the conductive coefficients a_{ij} decrease as the reciprocal of the length squared. If the same idea is applied to the convective coefficients it will be seen that the value of RC changes as the length changes and the convective rate coefficient, b_{ij} , changes as the $(\text{length})^{-1}$. As the model goes to full scale, the geometric properties become larger and thus the rate coefficients become smaller. Since the conductive values decrease as the reciprocal of the square of a length, where the convective values only decrease as the reciprocal of the length, the convective rate coefficients become dominant.

The quarter scale modeling conducted by Driels [Ref. 3], demonstrated that the value of the convective rate coefficient, b_{11} , from the tip section of the vane could be adjusted with minimal effect on the other unknown variables.

This was due to the ablative cooling of the tip due to the erosion of the tip area of the vane by the rocket exhaust. In the full scale experimental rocket firings there was minimal vane erosion as compared to the quarter scale vane tests. The energy in the full scale tests was not being carried away by the ablative cooling but was being transmitted into the vane. This was one of the primary reasons that led to making the value of the convective rate coefficient, b_{11} , a variable in the full scale model instead of the constant that was assumed based on the quarter scale model results.

CONCLUSIONS

- The four node quarter scale model cannot be directly scaled up to a full scale model due to the problems encountered with substantial jumps in the unknowns a_{56} and a_{65} , corresponding to the shaft and mount locations.
- The use of a constrained optimizer is necessary to hold the value of the unknowns to positive values and prevent the equation solver from experiencing an arithmetic fault or causing a fatal error because the energy balance equations became stiff.
- The value obtained in the quarter scale vane modeling process more adequately reflects the average convective heat transfer coefficient of the fin area where as the values obtained from the full scale modeling display a more accurate representation of the heat transfer process in the vane.
- The calculated profiles of Figure 3.17 indicate that as the vane size increases the convective heat transfer properties become the dominant player in the heat transfer process of the vane.
- The rise in the shaft and mount experimental temperature profiles is likely due to the rocket exhaust plume impinging directly on the shaft of the vane.
- The value of the convective rate coefficient, b_{11} , must be considered an unknown variable in the time-temperature matching process, due to the minimal amount of vane erosion and the lessening effect of ablative cooling in the full scale vane.

RECOMMENDATIONS FOR FURTHER STUDY

- It is recommended that further study modify the five node model by reducing it to three nodes and comparing the results to see if a single node in the fin area of the vane could produce an acceptable convective heat transfer coefficient.
- The full scale model needs to be able to take into account the effects of radiation from the casing of the rocket firing chamber. This may compensate for the non-linearities in the experimental temperature profiles.
- Vane erosion was evident in the quarter scale testing and some vane erosion occurred in the full scale testing. The model needs to be upgraded to include a component which can compensate for the erosion based on the metal content of the rocket exhaust.
- Since the convective rate coefficient cannot be considered a constant at the tip, a thermocouple attached to the tip of the vane to collect the experimental time-temperature data would greatly enhance the thermal model development.
- The process for the attachment of thermocouples needs to be addressed. The experimental data had glitches or spikes in the temperature profiles during various times of the rocket firing indicating that the thermocouple temporarily lost contact with the vane.

APPENDIX A. PID PROGRAM

The program contained is the FORTRAN code main program used in the vane modeling process. The programs name was changed to NODE5 to distinguish it from the PID program using the six node modeling approach.

Program NODE5

This program is the PID program for the five node vane model.

```
external temp
integer m,n,iparm(6),ibtype,ldfjac
parameter (m=183,n=5,ldfjac=m)
real*8 rparm(7),x(n),f(m),xjac(m,n),xg(n),
&      xlb(n),xub(n),xscale(n),fscale(m),
&      ssq,float,
&      a(5,5),b(5,5),u(5),t2(61),t3(61),t4(61),ys(5,61),
&      rho,k,cp,vt,vf,vs,atf,afs,ltf,lfs,sf,
&      atf1,atf2,atf3,vf1,vf2,vf3,ub1,ub2
intrinsic float
common/data1/a,b,u,t2,t3,t4,ys
```

Open files for data input/output

```
open(10,name='result.dat', status='new')
open(9,name='temp.mat',  status='new')
open(8,name='datam.dat', status='old')
open(7,name='input.dat', status='old')
```

read in experimental data

```
do i=1,61
    read(8,*) t2(i)
enddo
do i=1,61
    read(8,*) t3(i)
enddo
do i=1,61
    read(8,*) t4(i)
enddo
close(8)
```

read in input data

```
read(7,*)
read(7,*)
read(7,*)
read(7,*)
read(7,*) rho,k,cp
read(7,*)
read(7,*)
read(7,*) vt, vf1, vf2, vf3, vs
read(7,*)
read(7,*)
read(7,*) atf1, atf2, atf3, afs
read(7,*)
read(7,*)
read(7,*) ltf, lfs
read(7,*)
read(7,*) sf, ub1, ub2
close(7)
```

full scale data

```

r12=100.0*ltf/(k*atf1)
r23=100.0*ltf/(k*atf2)
r34=100.0*ltf/(k*atf3)
r45=100.0*lfs/(k*afs)
c1=rho*cp*vt*0.000001
c2=rho*cp*vf1*0.000001
c3=rho*cp*vf2*0.000001
c4=rho*cp*vf3*0.000001
c5=rho*cp*vs*0.000001

```

scaled data

```

r12=r12/sf
r23=r23/sf
r34=r34/sf
r45=r45/sf
c1=c1*sf**3
c2=c2*sf**3
c3=c3*sf**3
c4=c4*sf**3
c5=c5*sf**3

```

```
rf1=0.5
```

```

do i=1,5
u(i)=0.0
      do j=1,5
a(i,j)=0.0
b(i,j)=0.0
      enddo
enddo

```

```

a(1,2)=1/(c1*r12)
a(2,1)=1/(c2*r12)
a(2,3)=1/(c2*r23)
a(3,2)=1/(c3*r23)
a(3,4)=1/(c3*r34)
a(4,3)=1/(c4*r34)
a(4,5)=1/(c4*r45)
a(5,4)=1/(c5*r45)

```

```
b(1,1)=1/(c1*rf1)
```

```

write(6,*)'      a12          a21          a23          a32'
write(6,9002) a(1,2),a(2,1),a(2,3),a(3,2)
write(6,*)'      a34          a43          a45          a54'
write(6,9003) a(3,4),a(4,3),a(4,5),a(5,4)

```

```

002  format(5f9.3)
003  format(4f9.3)

```

```

a5g   =0.0
b(1,1)=0.0
b(2,2)=0.0
b(3,2)=0.0

```

```

b(4,2)=-0.0
a(1,1)=- (a(1,2)+b(1,1))
a(2,2)=- (a(2,1)+a(2,3)+b(2,2))
a(3,3)=- (a(3,2)+a(3,4)+b(3,2))
a(4,4)=- (a(4,3)+a(4,5)+b(4,2))
a(5,5)=- (a(5,4)+a5g)

```

```

u(1)=ub1
u(2)=ub2

```

```

xg(1)=a5g
xg(2)=b(1,1)
xg(3)=b(2,2)
xg(4)=b(3,2)
xg(5)=b(4,2)

```

```

set up parameters for DBCLSF call
do k=1,n
  xscale(k)=1.0
  xlb(k)=0.0001
end do

```

```

xub(1)=100.0
xub(2)=100.0
xub(3)=100.0
xub(4)=100.0
xub(5)=100.0

```

```

do l=1,m
  fscale(l)=1.0
end do

```

```

ibtype=0

```

```

& call dbclsf(temp,m,n,xg,ibtype,xlb,xub,xscale,fscale,
             iparm,rparm,x,f,xjac,ldfjac)

```

```

calculate unknown resistances and capacitances

```

```

a5g =x(1)
b(1,1)=x(2)
b(2,2)=x(3)
b(3,2)=x(4)
b(4,2)=x(5)

```

```

rf1 =1/(b(1,1)*c1)
rf22=1/(b(2,2)*c2)
rf23=1/(b(3,2)*c3)
rf24=1/(b(4,2)*c4)
r5g=1/(a5g*c5)

```

```

print and save results

```

```

write(6,*) '      a5g          b11          b22          b32'
write(6,9009) x(1),x(2),x(3),x(4),x(5)

```

```

000    format(5f12.4)

    write(10,*) 'a5g      b(1,1)      b(2,2)      b(3,2)'
    write(10,9000) x(1),x(2),x(3),x(4),x(5);
    write(10,*)
    write(10,*)
    write(10,*) 'rf1      rf22      rf23      rf24      r5g'
    write(10,9000) rf1rf22,rf23,rf24,r5g

    write the temp-time data for MATLAB analysis

    do i=1,61
    tt=0.0768*float(i)
    write(9,9001)tt,ys(2,i),ys(3,i),ys(4,i),t2(i),t3(i),t4(i)
    enddo
01    format(1f6.2,6f10.3)

    close(10)
    close(9)

    end

```

Subroutine TEMP (m,n,x,f)

This calculates the temperature-time history using the current parameters supplied by ZXSSQ called from PID. It calculates an error function returned to ZXSSQ based on the differences between predicted and observed temperature histories.

```

    integer maxparam, neq
    parameter(maxparam=50, neq=5)
    integer id0,istep,nout,m,n
    real*8 t,y(neq),a(neq,neq),b(neq,neq),u(neq),param(maxparam)
    real*8 tend,tol,fcn,float,a5g,ys(5,61),x(n),
&      f(m),t2(61),t3(61),t4(61)
    intrinsic float
    external fcnc,divprk,sset

    common/data1/a,b,u,t2,t3,t4,ys

    open(12,name='incoming.dat',status='new')

    a5g =x(1)
    b(1,1)=x(2)
    b(2,2)=x(3)
    b(3,2)=x(4)
    b(4,2)=x(5)

    write(6,8000)a5g,b(1,1),b(2,2),b(3,2),b(4,2)

00    format(5f12.4)

    a(1,1)=-(a(1,2)+b(1,1))
    a(2,2)=-(a(2,1)+a(2,3)+b(2,2))
    a(3,3)=-(a(3,2)+a(3,4)+b(3,2))
    a(4,4)=-(a(4,3)+a(4,5)+b(4,2))
    a(5,5)=-(a(5,4)+a5g)

```

```

t=0.0
do i=1,neq
y(i)=0.0
      do j=1,61
ys(i,j)=0.0
      enddo
enddo
tol=0.0005

call sset (maxparm, 0.0, param, 1)

id0=1
do istep=1,61
  tend=0.0768*float(istep)
  CALL DIVPRK (id0, neq, fcn, t, tend, tol, param, y)
  do i=1,5
ys(i,istep)=y(i)
  enddo
enddo

Final call to release workspace

id0=3
call divprk (id0,neq,fcn,t,tend,tol,param,y)

calculate error functions

do i=1,61
f(i)=ys(2,i)-t2(i)
f(i+61)=ys(3,i)-t3(i)
f(i+122)=ys(4,i)-t4(i)
enddo

print out rms error
ssqr=0.0
do i=1,122
ssqr=ssqr+f(i)*f(i)
enddo
ssqr=ssqr/122
xer=sqrt(ssqr)
write(6,*) xer
write(12,*) xer
return
end

```

```

subroutine fcn(neq,t,y,yprime)

integer neq
real*8 t,y(neq),yprime(neq)
real*8 a(5,5),b(5,5),u(5),d,ys(5,61)
common/data1/a,b,u,t2,t3,t4,ys

thrust profile simulation as step input

if (t.gt.0.2) then
  d=1.0

```

```
else
    d=0.0
end if

do i=1,neq
yprime(i)=0.0
    do j=1,neq
        yprime(i)=yprime(i)+a(i,j)*y(j)+b(i,j)*u(j)*d
    enddo
enddo

return
end
```

APPENDIX B. GEOMETRIC AND PHYSICAL DATA FILE

The file in this appendix, INPUT.DAT, is the file containing the geometric and physical properties of the vane. The values are listed in full scale.

NAWC INVERSE HEAT TRANSFER PROGRAM. INPUT DATA FOR FULL SCALE

Material properties: rho (kg/m^3), k (w/m-deg k), Cp (J/kg deg k)
18310.0, 173.0, 146.0

Vol (tip, cm^3), Vol (fin, cm^3), Vol (shaft, cm^3)
2.6 12.86 17.15 21.44 23.0

X-section areas: tip-fin (cm^2) fin-shaft (cm^2)
5.6 5.9 6.2 5.2

Conductive path tip-fin (cm) fin-shaft (cm)
2.774 5.0

Scale factor:
1.00 2670 2570

APPENDIX C. TPROFILE - FORWARD MODELING PROGRAM

The program this appendix is the forward modeling program based on a modification of the main program, PID, and used to illustrate the effects of the unknown variables on the resultant time temperature profiles.

Program TPROFILE

This is a forward model program developed from the program PID by removing the optimizer.

```
external temp
integer m,n,ixjac,nsig,maxfn,iopt,infer,ier
parameter (m=183,n=5)
real*8 parm(6),x(n),f(m),xjac(m,n),xjtj((n+1)*n/2),
&      work(5*n+2*m+((n+1)*n/2)),
&      eps,delta,ssq,ub1,ub2,
&      a(5,5),b(5,5),u(5),t2(61),t3(61),t4(61),ys(5,61),
&      rho,k,cp,vt,vf,vs,atf,afs,ltf,lfs,sf
common/datal/a,b,u,t2,t3,t4,ys
```

Open files for data input/output

```
open(10,name='result.dat', status='new')
open(9,name='temp.mat',  status='new')
open(8,name='datam.dat', status='old')
open(7,name='input.dat', status='old')
```

```
do i=1,61
    read(8,*) t2(i)
enddo
do i=1,61
    read(8,*) t3(i)
enddo
do i=1,61
    read(8,*) t4(i)
enddo
close(8)
```

read in input data

```
read(7,*)
read(7,*)
read(7,*)
read(7,*)
read(7,*) rho,k,cp
read(7,*)
read(7,*)
read(7,*) vt, vf1, vf2, vf3, vs
read(7,*)
read(7,*)
read(7,*) atf1, atf2, atf3, afs
read(7,*)
read(7,*)
read(7,*) ltf, lfs
read(7,*)
read(7,*) sf, ub1, ub2
close(7)
```

initial conditions

full scale data

```

r12=100.0*ltf/(k*atf1)
r23=100.0*ltf/(k*atf2)
r34=100.0*ltf/(k*atf3)
r45=100.0*lfs/(k*afs)
c1=rho*cp*vt*0.000001
c2=rho*cp*vf1*0.000001
c3=rho*cp*vf2*0.000001
c4=rho*cp*vf3*0.000001
c5=rho*cp*vs*0.000001

```

```

write(6,*)r12,r23,r34,r45,c1,c2,c3,c4,c5

```

```

scaled data

```

```

r12=r12/sf
r23=r23/sf
r34=r34/sf
r45=r45/sf
c1=c1*sf**3
c2=c2*sf**3
c3=c3*sf**3
c4=c4*sf**3
c5=c5*sf**3

```

```

rf1=0.076318

```

```

do i=1,5
u(i)=0.0
      do j=1,5
a(i,j)=0.0
b(i,j)=0.0
      enddo
enddo

```

```

a(1,2)=1/(c1*r12)
a(2,1)=1/(c2*r12)
a(2,3)=1/(c2*r23)
a(3,2)=1/(c3*r23)
a(3,4)=1/(c3*r34)
a(4,3)=1/(c4*r34)
a(4,5)=1/(c4*r45)
a(5,4)=1/(c5*r45)

```

```

b(1,1)=1/(c1*rf1)

```

```

write(6,*)'      a12      a21      a23      a32      b11'
write(6,9002) a(1,2),a(2,1),a(2,3),a(3,2),b(1,1)
write(6,*)'      a34      a43      a45      a54'
write(6,9003) a(3,4),a(4,3),a(4,5),a(5,4)

```

```

)02  format(5f9.3)
)03  format(4f9.3)

```

```

a5g   =0.0001
b(1,1)=300.0
b(2,2)=1.339
b(3,2)=0.5657
b(4,2)=0.0037

```

```

a(1,1)=- (a(1,2)+a(1,3)+b(1,1))
a(2,2)=- (a(2,1)+a(2,3)+b(2,2))
a(3,3)=- (a(3,2)+a(3,4)+b(3,2))
a(4,4)=- (a(4,3)+a(4,5)+b(4,2))
a(5,5)=- (a(5,4)+a5g)

```

```

u(1)=ub1
u(2)=ub2

```

```

x(1)=a5g
x(2)=b(1,1)
x(3)=b(2,2)
x(4)=b(3,2)
x(5)=b(4,2)

```

```

call temp(x,m,n,f)

```

write the temp-time data for MATLAB analysis

```

do i=1,61
tt=.0768*float(i)
write(9,9001)tt,ys(2,i),ys(3,i),ys(4,i),t2(i),t3(i),t4(i)
enddo
01  format(1f6.2,6f10.3)

close(10)
close(9)
end

```

Subroutine TEMP (x,m,n,f)

This calculates the temperature-time history using the current parameters supplied by ZXSSQ called from PID. It calculates an error function returned to ZXSSQ based on the differences between predicted and observed temperature histories.

```

integer maxparam, neq
parameter(maxparam=50, neq=5)
integer id0, istep, nout, m, n
real*8 t, y(neq), a(neq, neq), b(neq, neq), u(neq), param(maxparam)
real*8 tend, tol, fcn, float, a5g, ys(5, 61), x(n),
&      f(m), t2(61), t3(61), t4(61)
intrinsic float
external fcn, divprk, sset

```

```

common/data1/a,b,u,t2,t3,t4,ys

```

```

a5g    =x(1)
b(1,1)=x(2)
b(2,2)=x(3)
b(3,2)=x(4)
b(4,2)=x(5)

```

```

a(1,1)=- (a(1,2)+b(1,1))
a(2,2)=- (a(2,1)+a(2,3)+b(2,2))
a(3,3)=- (a(3,2)+a(3,4)+b(3,2))
a(4,4)=- (a(4,3)+a(4,5)+b(4,2))

```

Set initial conditions

```
t=0.0
do i=1,neq
y(i)=0.0
      do j=1,61
        ys(i,j)=0.0
      enddo
enddo
tol=0.0005

call sset (maxparm, 0.0, param, 1)

param(4)=2000

id0=1
do istep=1,61
  tend=0.0768*float(istep)
  CALL DIVPRK (id0, neq, fcn, t, tend, tol, param, y)
  do i=1,5
    ys(i,istep)=y(i)
  enddo
enddo

Final call to release workspace

id0=3
call divprk (id0,neq,fcn,t,tend,tol,param,y)

do i=1,61
f(i)=ys(2,i)-t2(i)
f(i)=ys(3,i)-t3(i)
f(i)=ys(4,i)-t4(i)
enddo

ssqr=0.0
do i=1,61
ssqr=ssqr+f(i)*f(i)
enddo
ssqr=ssqr/61
xer=sqrt(ssqr)
write(6,*) xer
return
end
```

subroutine fcn(neq,t,y,yprime)

```
integer neq
real*8 t,y(neq),yprime(neq)
real*8 a(5,5),b(5,5),u(5),d,ys(5,61)
common/datal/a,b,u,t2,t3,t4,ys

if (t.gt.0.2) then
  d=1.0
else
  d=0.0
```

```
end if
.00 do i=1,neq
yprime(i)=0.0
      do j=1,neq
yprime(i)=yprime(i)+a(i,j)*y(j)+b(i,j)*u(j)*d
      enddo
enddo

return
end
```

APPENDIX D. CONSTRAINED OPTIMIZER - DBCLSF

This optimizer is the constrained optimizer which allows the user to place constraints on the range and values of the unknowns. The constraints may be a set of boundary limits or restraining the unknown variables to either being negative or positive.

BCLSJ/DBCLSJ (Single/Double precision)

Purpose: Solve a nonlinear least squares problem subject to bounds on the variables using a modified Levenberg-Marquardt algorithm and a user-supplied Jacobian.

Usage: CALL BCLSJ (FCN, JAC, M, N, XGUESS, IBTYPE, XLB, XUB, XSCALE, FSCALE, IPARAM, RPARAM, X, FVEC, FJAC, LDFJAC)

Arguments

- FCN** - User-supplied SUBROUTINE to evaluate the function to be minimized. The usage is
CALL FCN (M, N, X, F), where
M - Length of F. (Input)
N - Length of X. (Input)
X - The point at which the function is evaluated.
(Input)
X should not be changed by FCN.
F - The computed function at the point X.
(Output)
FCN must be declared EXTERNAL in the calling program.
- JAC** - User-supplied SUBROUTINE to evaluate the Jacobian at a point X. The usage is
CALL JAC (M, N, X, FJAC, LDFJAC), where
M - Length of F. (Input)
N - Length of X. (Input)
X - The point at which the function is evaluated.
(Input)
X should not be changed by FCN.
FJAC - The computed M by N Jacobian at the point X.
(Output)
LDFJAC - Leading dimension of FJAC. (Input)
JAC must be declared EXTERNAL in the calling program.
- M** - Number of functions. (Input)
N - Number of variables. (Input)
XGUESS - Vector of length N containing the initial guess. (Input)
IBTYPE - Scalar indicating the types of bounds on variables.
(Input)
IBTYPE Action
0 User will supply all the bounds.
1 All variables are nonnegative.
2 All variables are nonpositive.
3 User supplies only the bounds on 1st variable.

all other variables will have the same bounds.

- XLB** - Vector of length N containing the lower bounds on variables. (Input, if $IBTYPE = 0$; output, if $IBTYPE = 1$ or 2 ; input/output, if $IBTYPE = 3$)
- XUB** - Vector of length N containing the upper bounds on variables. (Input, if $IBTYPE = 0$; output, if $IBTYPE = 1$ or 2 ; input/output, if $IBTYPE = 3$)
- XSCALE** - Vector of length N containing the diagonal scaling matrix for the variables. (Input)
In the absence of other information, set all entries to 1.0.
- FSCALE** - Vector of length M containing the diagonal scaling matrix for the functions. (Input)
In the absence of other information, set all entries to 1.0.
- IPARAM** - Parameter vector of length 6. (Input/Output)
See Remarks.
- RPARAM** - Parameters vector of length 7. (Input/Output)
See Remarks.
- X** - Vector of length N containing the approximate solution. (Output)
- FVEC** - Vector of length M containing the residuals at the approximate solution. (Output)
- FJAC** - M by N matrix containing a finite difference approximate Jacobian at the approximate solution. (Output)
- LDFJAC** - Leading dimension of **FJAC** exactly as specified in the dimension statement of the calling program. (Input)

Remarks

1. Automatic workspace usage is

BCLSJ $14 \cdot N + 2 \cdot M - 1$ units, or

DBCLSJ $26 \cdot N + 4 \cdot M - 2$ units.

Workspace may be explicitly provided, if desired, by use of **B2LSJ/DB2LSJ**. The reference is

CALL **B2LSJ** (**FCN**, **JAC**, N , N , **XGUESS**, **IBTYPE**, **XLB**, **XUB**,
XSCALE, **FSCALE**, **IPARAM**, **RPARAM**, **X**, **FVEC**,
FJAC, **LDFJAC**, **WK**, **IWK**)

The additional arguments are as follows:

- WK** - Work vector of length $12 \cdot N + 2 \cdot M - 1$. **WK** contains the following information on output:
The second N locations contain the last step taken.
The third N locations contain the last Gauss-Newton step.
The fourth N locations contain an estimate of the gradient at the solution.

IWK - Work vector of length $2*N$ containing the permutations used in the QR factorization of the Jacobian at the solution.

2. Informational errors

Type Code

- 3 1 Both the actual and predicted relative reductions in the function are less than or equal to the relative function convergence tolerance.
- 4 2 The iterates appear to be converging to a noncritical point.
- 4 3 Maximum number of iterations exceeded.
- 4 4 Maximum number of function evaluations exceeded.
- 4 5 Five consecutive steps have been taken with the maximum step length.
- 4 6 Maximum number of Jacobian evaluations exceeded.

3. The first stopping criterion for BCLSJ occurs when the norm of the function is less than the absolute function tolerance. The second stopping criterion occurs when the norm of the scaled gradient is less than the given gradient tolerance. The third stopping criterion for BCLSJ occurs when the scaled distance between the last two steps is less than the step tolerance.

4. If nondefault parameters are desired for IPARAM or RPARAM, then U4LSF is called and the corresponding parameters are set to the desired value before calling the optimization program. Otherwise, if the default parameters are desired, then set IPARAM(1) to zero and call the optimization program omitting the call to U4LSF. The call to U4LSF would be as follows:

```
CALL U4LSF (IPARAM, RPARAM).
```

The following is a list of the parameters and the default values:

IPARAM - Integer vector of length 6.

IPARAM(1) = Initialization flag. (0)

IPARAM(2) = Number of good digits in the function.
(Machine dependent)

IPARAM(3) = Maximum number of iterations. (100)

IPARAM(4) = Maximum number of function evaluations. (400)

IPARAM(5) = Maximum number of Jacobian evaluations. (100)

IPARAM(6) = Internal variable scaling flag. (1)

If IPARAM(6) = 1 the values for XSCALE are set internally.

RPARAM - Real vector of length 7.

RPARAM(1) = Scaled gradient tolerance. (eps**(2/3))
 RPARAM(2) = Scaled step tolerance. (eps**(2/3))
 RPARAM(3) = Relative function tolerance.
 (MAX(1.0E-10,eps**(2/3)))
 RPARAM(4) = Absolute function tolerance.
 (MAX(1.0E-10,eps**(2/3)))
 RPARAM(5) = False convergence tolerance. (100*eps)
 RPARAM(6) = Maximum allowable step size.
 (1000*MAX(TOL1,TOL2)) where,
 TOL1 = SQRT(sum of (XSCALE(I)*XGUESS(I))**2)
 for I = 1,...,N
 TOL2 = 2-norm of XSCALE.
 RPARAM(7) = Size of initial trust region radius.
 (Based on the initial scaled Cauchy step)

eps is machine epsilon.

If double precision is desired, then DU4LSF is called and RPARAM is declared double precision.

Keywords: Levenberg-Marquardt; Trust region

Algorithm

BCLSJ uses a modified Levenberg-Marquardt method and an active set strategy to solve nonlinear least squares problems subject to simple bounds on the variables. The problem is stated as follows:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} F(x)^T F(x) = \frac{1}{2} \sum_{i=1}^m f_i(x)^2$$

subject to $l \leq x \leq u$.

where $m \geq n$, $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$, and $f_i(x)$ is the i -th component function of $F(x)$. From a given starting point, an active set IA, which contains the indices of the variables at their bounds, is built. A variable is called a 'free variable' if it is not in the active set. The routine then computes the search direction for the free variables according to the formula

$$d = -(J^T J + \mu I)^{-1} J^T F,$$

where μ is the Levenberg-Marquardt parameter, $F = F(x)$, and J is the Jacobian with respect to the free variables. The search direction for the variables in IA is set to zero. The trust region approach discussed by Dennis and Schnabel (1983) is used to find the new point. Finally, the optimality conditions are checked. The conditions are:

$$\|g(x_i)\| \leq \epsilon, \quad l_i < x_i < u_i$$

$$g(x_i) < 0, \quad x_i = u_i$$

$$g(x_i) > 0, \quad x_i = l_i,$$

where ϵ is a gradient tolerance. This process is repeated until the optimality criterion is achieved.

The active set is changed only when a free variable hits its bounds during an iteration, or the optimality condition is met for the free variables but not for all variables in IA, the active set. In the latter case, a variable which violates the optimality condition will be dropped out of IA. For more detail on the Levenberg-Marquardt method, see Levenberg (1944), or Marquardt (1963). For more detailed information on active set strategy, see Gill and Murray (1976).

Example

The nonlinear least squares problem

$$\min_{x \in \mathbb{R}^2} \frac{1}{2} \sum_{i=1}^2 f_i(x)^2$$

$$\text{subject to } -2 \leq x_1 \leq 0.5 \\ -1 \leq x_2 \leq 2.$$

where $f_1(x) = 10(x_2 - x_1^2)$, and $f_2(x) = (1 - x_1)$ is solved with an initial guess $(-1.2, 1.0)$, and default values for parameters.

```

C                                     Declaration of variables
      INTEGER      LDFJAC, M, N
      PARAMETER    (LDFJAC=2, M=2, N=2)

C
      INTEGER      IPARAM(7), ITP, NOUT
      REAL         FJAC(LDFJAC,N), FSCALE(M), FVEC(N), ROSBCK, ROSJAC,
      *            RPARAM(7), X(N), XGUESS(N), XLB(N), XS(N), XUB(N)
      EXTERNAL     BCLSJ, ROSBCK, ROSJAC, UMACH

C                                     Compute the least squares for the
C                                     Rosenbrock function.
      DATA XGUESS/-1.2E0, 1.0E0/, XS/2=1.0E0/, FSCALE/2=1.0E0/
      DATA XLB/-2.0E0, -1.0E0/, XUB/0.5E0, 2.0E0/

C                                     All the bounds are provided
      ITP = 0

C                                     Default parameters are used
      IPARAM(1) = 0

C
      CALL BCLSJ (ROSBCK, ROSJAC, M, N, XGUESS, ITP, XLB, XUB, XS,
      *          FSCALE, IPARAM, RPARAM, X, FVEC, FJAC, LDFJAC)

C                                     Print results
      CALL UMACH (2, NOUT)
      WRITE (NOUT,99999) X, FVEC, IPARAM(3), IPARAM(4)

C
99999 FORMAT (' The solution is ', 2F9.4, '//, ' The function ',

```

```

&      'evaluated at the solution is ', /, 18X, 2F9.4, //,
&      ' The number of iterations is ', 10X, I3, /, ' The ',
&      'number of function evaluations is ', I3, /)
      END
C
      SUBROUTINE ROSBCK (M, N, X, F)
      INTEGER      M, N
      REAL          X(N), F(M)
C
      F(1) = 1.0E1-(X(2)-X(1)-X(1))
      F(2) = 1.0E0 - X(1)
      RETURN
      END
C
      SUBROUTINE ROSJAC (M, N, X, FJAC, LDFJAC)
      INTEGER      M, N, LDFJAC
      REAL          X(N), FJAC(LDFJAC,N)
C
      FJAC(1,1) = -20.0E0*X(1)
      FJAC(2,1) = -1.0E0
      FJAC(1,2) = 10.0E0
      FJAC(2,2) = 0.0E0
      RETURN
      END

```

Output

```

The solution is      .5000      .2500

The function evaluated at the solution is
                    .0000      .5000

The number of iterations is          13
The number of function evaluations is 21

```

References

- Dennis, J. E., Jr., and Robert B. Schnabel (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, New Jersey.
- Gill, Philip E., and Walter Murray (1976). *Minimization subject to bounds on the variables*. NPL Report NAC 72, National Physical Laboratory, England.
- Levenberg, K. (1944). A method for the solution of certain problems in least squares. *Quarterly of Applied Mathematics*, 2, 164-168.
- Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11, 431-441.

APPENDIX E. ZXSSQ OPTIMIZER

This optimizer routine was the initial optimizing program in the vane modeling process using the IMSL subroutines. It provides no way of constraining the values of the unknown variables.

IMSL ROUTINE NAME - ZXSSQ

PURPOSE - MINIMUM OF THE SUM OF SQUARES OF M FUNCTIONS
IN N VARIABLES USING A FINITE DIFFERENCE
LEVENBERG-MARQUARDT ALGORITHM

USAGE - CALL ZXSSQ(FUNC,M,N,NSIG,EPS,DELTA,MAXFN,IOPT,
PARM,X,SSQ,F,XJAC,IXJAC,XJTJ,WORK,INFER,IER)

ARGUMENTS

FUNC	- A USER SUPPLIED SUBROUTINE WHICH CALCULATES THE RESIDUAL VECTOR $F(1), F(2), \dots, F(M)$ FOR GIVEN PARAMETER VALUES $X(1), X(2), \dots, X(N)$. THE CALLING SEQUENCE HAS THE FOLLOWING FORM CALL FUNC(X,M,N,F) WHERE X IS A VECTOR OF LENGTH N AND F IS A VECTOR OF LENGTH M. FUNC MUST APPEAR IN AN EXTERNAL STATEMENT IN THE CALLING PROGRAM. FUNC MUST NOT ALTER THE VALUES OF $X(I), I=1, \dots, N, M$, OR N.
M	- THE NUMBER OF RESIDUALS OR OBSERVATIONS (INPUT)
N	- THE NUMBER OF UNKNOWN PARAMETERS (INPUT).
NSIG	- FIRST CONVERGENCE CRITERION. (INPUT) CONVERGENCE CONDITION SATISFIED IF ON TWO SUCCESSIVE ITERATIONS, THE PARAMETER ESTIMATES AGREE, COMPONENT BY COMPONENT, TO NSIG DIGITS.
EPS	- SECOND CONVERGENCE CRITERION. (INPUT) CONVERGENCE CONDITION SATISFIED IF, ON TWO SUCCESSIVE ITERATIONS THE RESIDUAL SUM OF SQUARES ESTIMATES HAVE RELATIVE DIFFERENCE LESS THAN OR EQUAL TO EPS. EPS MAY BE SET TO ZERO.
DELTA	- THIRD CONVERGENCE CRITERION. (INPUT) CONVERGENCE CONDITION SATISFIED IF THE (EUCLIDEAN) NORM OF THE APPROXIMATE GRADIENT IS LESS THAN OR EQUAL TO DELTA. DELTA MAY BE SET TO ZERO. NOTE, THE ITERATION IS TERMINATED, AND CONVERGENCE IS CONSIDERED ACHIEVED. IF ANY ONE OF THE THREE CONDITIONS IS SATISFIED.
MAXFN	- INPUT MAXIMUM NUMBER OF FUNCTION EVALUATIONS (I.E., CALLS TO SUBROUTINE FUNC) ALLOWED. THE ACTUAL NUMBER OF CALLS TO FUNC MAY EXCEED MAXFN SLIGHTLY.
IOPT	- INPUT OPTIONS PARAMETER. IOPT=0 IMPLIES BROWN'S ALGORITHM WITHOUT STRICT DESCENT IS DESIRED. IOPT=1 IMPLIES STRICT DESCENT AND DEFAULT VALUES FOR INPUT VECTOR PARM ARE DESIRED. IOPT=2 IMPLIES STRICT DESCENT IS DESIRED WITH USER PARAMETER CHOICES IN INPUT VECTOR PARM.
PARM	- INPUT VECTOR OF LENGTH 4 USED ONLY FOR IOPT EQUAL TWO. PARM(1) CONTAINS, WHEN

ZEROS AND EXTREMA

ZXSSQ-1

- I-1, THE INITIAL VALUE OF THE MARQUARDT PARAMETER USED TO SCALE THE DIAGONAL OF THE APPROXIMATE HESSIAN MATRIX, XJTJ, BY THE FACTOR $(1.0 + \text{PARM}(1))$. A SMALL VALUE GIVES A NEWTON STEP, WHILE A LARGE VALUE GIVES A STEEPEST DESCENT STEP. THE DEFAULT VALUE FOR PARM(1) IS 0.01.
- I-2, THE SCALING FACTOR USED TO MODIFY THE MARQUARDT PARAMETER, WHICH IS DECREASED BY PARM(2) AFTER AN IMMEDIATELY SUCCESSFUL DESCENT DIRECTION, AND INCREASED BY THE SQUARE OF PARM(2) IF NOT. PARM(2) MUST BE GREATER THAN ONE, AND TWO IS DEFAULT.
- I-3, AN UPPER BOUND FOR INCREASING THE MARQUARDT PARAMETER. THE SEARCH FOR A DESCENT POINT IS ABANDONED IF PARM(3) IS EXCEEDED. PARM(3) GREATER THAN 100.0 IS RECOMMENDED. DEFAULT IS 120.0.
- I-4, VALUE FOR INDICATING WHEN CENTRAL RATHER THAN FORWARD DIFFERENCING IS TO BE USED FOR CALCULATING THE JACOBIAN. THE SWITCH IS MADE WHEN THE NORM OF THE GRADIENT OF THE SUM OF SQUARES FUNCTION BECOMES SMALLER THAN PARM(4). CENTRAL DIFFERENCING IS GOOD IN THE VICINITY OF THE SOLUTION, SO PARM(4) SHOULD BE SMALL. THE DEFAULT VALUE IS 0.10.
- X - VECTOR OF LENGTH N CONTAINING PARAMETER VALUES.
ON INPUT, X SHOULD CONTAIN THE INITIAL ESTIMATE OF THE LOCATION OF THE MINIMUM.
ON OUTPUT, X CONTAINS THE FINAL ESTIMATE OF THE LOCATION OF THE MINIMUM.
- SSQ - OUTPUT SCALAR WHICH IS SET TO THE RESIDUAL SUMS OF SQUARES, $F(1)^2 + \dots + F(M)^2$, FOR THE FINAL PARAMETER ESTIMATES.
- F - OUTPUT VECTOR OF LENGTH N CONTAINING THE RESIDUALS FOR THE FINAL PARAMETER ESTIMATES.
- XJAC - OUTPUT M BY N MATRIX CONTAINING THE APPROXIMATE JACOBIAN AT THE OUTPUT VECTOR X.
- IXJAC - INPUT ROW DIMENSION OF MATRIX XJAC EXACTLY AS SPECIFIED IN THE DIMENSION STATEMENT IN THE CALLING PROGRAM.
- XJTJ - OUTPUT VECTOR OF LENGTH $(N+1)*N/2$ CONTAINING THE N BY N MATRIX (XJAC-TRANPOSED) * (XJAC) IN SYMMETRIC STORAGE MODE.
- WORK - WORK VECTOR OF LENGTH $5*N + 2*M + (N+1)*N/2$.
ON OUTPUT, WORK(I) CONTAINS FOR
I-1, THE NORM OF THE GRADIENT DESCRIBED UNDER INPUT PARAMETERS DELTA AND PARM(4).
I-2, THE NUMBER OF FUNCTION EVALUATIONS REQUIRED DURING THE WORK(5) ITERATIONS.
I-3, THE ESTIMATED NUMBER OF SIGNIFICANT DIGITS IN OUTPUT VECTOR X.
I-4, THE FINAL VALUE OF THE MARQUARDT SCALING PARAMETER DESCRIBED UNDER PARM(1).

I=5, THE NUMBER OF ITERATIONS (I.E., CHANGES TO THE X VECTOR) PERFORMED.

SEE PROGRAMMING NOTES FOR DESCRIPTION OF THE LATTER ELEMENTS OF WORK.

INFER - AN INTEGER THAT IS SET, ON OUTPUT, TO INDICATE WHICH CONVERGENCE CRITERION WAS SATISFIED.

INFER = 0 INDICATES THAT CONVERGENCE FAILED. IER GIVES FURTHER EXPLANATION.

INFER = 1 INDICATES THAT THE FIRST CRITERION WAS SATISFIED.

INFER = 2 INDICATES THAT THE SECOND CRITERION WAS SATISFIED.

INFER = 4 INDICATES THAT THE THIRD CRITERION WAS SATISFIED.

IF MORE THAN ONE OF THE CONVERGENCE CRITERIA WERE SATISFIED ON THE FINAL ITERATION, **INFER** CONTAINS THE CORRESPONDING SUM. (E.G., **INFER** = 3 IMPLIES FIRST AND SECOND CRITERIA SATISFIED SIMULTANEOUSLY).

IER - ERROR PARAMETER (OUTPUT)
TERMINAL ERROR

IER=129 IMPLIES A SINGULARITY WAS DETECTED IN THE JACOBIAN AND RECOVERY FAILED.

IER=130 IMPLIES AT LEAST ONE OF M, N, IOPT, PARM(1), OR PARM(2) WAS SPECIFIED INCORRECTLY.

IER=132 IMPLIES THAT AFTER A SUCCESSFUL RECOVERY FROM A SINGULAR JACOBIAN, THE VECTOR X HAS CYCLED BACK TO THE FIRST SINGULARITY.

IER=133 IMPLIES THAT MAXFN WAS EXCEEDED.
WARNING ERROR

IER=38 IMPLIES THAT THE JACOBIAN IS ZERO. THE SOLUTION X IS A STATIONARY POINT.

IER=39 IMPLIES THAT THE MARQUARDT PARAMETER EXCEEDED PARM(3). THIS USUALLY MEANS THAT THE REQUESTED ACCURACY WAS NOT ACHIEVED.

PRECISION

- SINGLE AND DOUBLE

Algorithm

ZXSSQ is a finite difference, Levenberg-Marquardt routine for solving nonlinear least squares problems. The problem is stated as follows:

given M nonlinear functions f_1, f_2, \dots, f_M of a vector parameter \underline{x} ,

minimize $f_1(\underline{x})^2 + f_2(\underline{x})^2 + \dots + f_M(\underline{x})^2$
over \underline{x}

where $\underline{x} = (x_1, x_2, \dots, x_N)$ is a vector of N parameters to be estimated.

When fitting a nonlinear model to data, the functions f_i should be defined as follows:

$$f_i(\underline{x}) = y_i - g(\underline{x}; v_i^1) \quad i=1, 2, \dots, M \quad (\text{i.e., the residuals})$$

where y_i is the i -th observation of the dependent variable,

$v_i^1 = (v_{i1}^1, v_{i2}^1, \dots, v_{iN}^1)$ is a vector containing the i -th observation of the NV independent variables, and

g is the function defining the nonlinear model.

ZXSSQ is based on a modification of the Levenberg-Marquardt algorithm which eliminates the need for explicit derivatives.

Let \underline{x}^0 be an initial estimate of \underline{x} . A sequence of approximations to the minimum point is generated by

$$\underline{x}^{n+1} = \underline{x}^n - [\alpha_n D_n + J_n^T J_n]^{-1} J_n^T f(\underline{x}^n),$$

where J_n is the numerical Jacobian matrix evaluated at \underline{x}^n

D_n is a diagonal matrix equal to the diagonal of $J_n^T J_n$
for $IOPT \neq 0$.

α_n is a positive scaling constant (Marquardt parameter)

When forward differences are used, the Jacobian is calculated by

$$\frac{1}{h_j} [f_i(\underline{x} + h_j \underline{u}_j) - f_i(\underline{x})]$$

where \underline{u}_j is the j -th unit vector and $h_j = \max(|x_j|, 0.1) \cdot \text{eps}^{1/2}$ (see scaling comments in Programming Note 2), where eps is the relative precision of floating point arithmetic. For central differences,

$$\frac{1}{2h_j} [f_i(\underline{x} + h_j \underline{u}_j) - f_i(\underline{x} - h_j \underline{u}_j)]$$

is used. To minimize the number of function evaluations required for nonzero $IOPT$, a rank one update to the Jacobian matrix is used where appropriate:

$$J_{n+1} = J_n + \frac{1}{\delta^T \delta} [f(x^{n+1}) - f(x^n) - J_n \delta] \delta^T$$

where $\delta = x^{n+1} - x^n$.

See references:

1. Brown, K. M., and Dennis, J. E., "Derivative free analogues of the Levenberg-Marquardt and Gauss algorithms for nonlinear least squares approximations", Numerische Mathematik, 18, 1972, 289-297.
2. Brown, K. M., "Computer oriented methods for fitting tabular data in the linear and nonlinear least squares sense", Department of Computer, Information, and Control Sciences, TR No. 72-13, University of Minnesota.
3. Levenberg, K., "A method for the solution of certain non-linear problems in least squares", Quart. Appl. Math., 2, 1944, 164-168.
4. Marquardt, D. W., "An algorithm for least-squares estimation of nonlinear parameters", J. SIAM, 11(2)1963.

Programming Notes

1. Accuracy in evaluating the functions f_i is critical when highly accurate parameter values, x_j , are required (i.e., when NSIG is greater than 3 for single precision). In these cases, it is advisable to evaluate the functions in precision greater than the working precision (e.g., for single precision ZXSSQ, double precision can be used).
2. In reference to significant digit tests for the x vector and the sum of squares, leading zeroes to the right of the decimal point are counted. For example, both 123.45 and 0.00123 have five significant digits. Scaling the x vector in the functions f_i may be required if several of the parameters x_j are much less than 1.0 to obtain the same number of significant digits for each of the x_j .
3. For IOPT \neq 0, strict descent is enforced by increasing the Marquardt parameter α_n (see PARM(1)) by the factor $\text{PARM}(2)^2$ until a smaller sum of squares is obtained. If a smaller SQQ is obtained immediately during iteration, α_n is decreased by the factor PARM(2). α_{n+1} is also modified by the ratio of the norm of the gradient $2||J^T r||$ for two successive iterations in the range $(\text{PARM}(2)^{-1}, \text{PARM}(2))$, decreasing α_{n+1} as the gradient decreases, and increasing α_{n+1} as the gradient increases.

4. For IOPT=0, a formula is used to calculate α_n , and no check is made to see that SSQ has been reduced. The successive iterates, x^n , are accepted even if SSQ increases. IOPT=0 seems to be most effective when SSQ has a minimum near zero. In other cases and when IOPT=0 fails, it is advisable to use IOPT=1 or 2.
5. The Jacobian has typical element $\partial f_i / \partial x_j$, for $i=1,2,\dots,M$ and $j=1,2,\dots,N$. The Hessian has typical element $\partial^2(\text{SSQ}) / \partial x_j \partial x_k$, for $j,k=1,2,\dots,N$. Note that $\text{SSQ} = f^T f$. For small residuals f_i , the output matrix $XJTJ$ is a good approximation to the Hessian times one half.
6. For normal completion (IER=0), a Jacobian calculated by forward or central differences (not a rank one update version) is returned in ZJAC. Also, no convergence tests are performed while the rank one approximation is being used.
7. If a terminal error other than IER=130 is returned, all output parameters contain the values indicated for the iteration during which the terminal error occurred. In particular, the five values returned in vector WORK should be inspected whether a terminal error was returned or not. If IER=131 is returned, WORK(3) will overestimate the number of significant digits in x by 2 or 3 digits.
8. Automatic recovery from singularities in the Jacobian (i.e., an entire column is zero) is provided.
9. Theoretically, the norm of the gradient, $2||J^T f||$, should be zero at the least squares solution, so that WORK(1) should be inspected before a solution is accepted. Usually, values less than 10^{-4} are acceptable, but this number is also dependent on the scale of the functions. The gradient times one half is located in output vector WORK(J+K), where $K = \text{MAX}((N+1)*N/2, 5)$ and $J=1,2,\dots,N$. $\text{WORK}_{j+k} = \partial(\text{SSQ}) / \partial x_j$.

Example

Suppose the nonlinear model $g(x; v^1) = x_1 \cdot v^1 + \sin(x_2 v^1)$ is to be fitted to 5 data points $y_i = v^1 + \sin(v^1) + (0.1) \cdot (-1)^i$ where $v^1 = 1$, for $i=1,2,\dots,5$. Then $f_i = y_i - g(x; v^1)$ and for

APPENDIX F. DATA PROCESSING PROGRAMS

The programs in this appendix, TRANS.FOR and PRO.M, are used to process the experimental data for use in the modeling program. TRANS.FOR is a FORTRAN code program which averages the values of the experimental data and PRO.M is a MATLAB program which normalizes the resultant data, from TRANS.FOR, to ambient and converts the temperatures from degree Fahrenheit to degrees Kelvin.

Program TRANS

```
c
c      This program translates data into free format and smooths
c      the data with a moving window.
c
c      real*8 x(82),y(82),x1(61),y1(61)
c
c      open(8,name='pre3.dat',status='old')
c      open(9,name='tv1.mat',status='new')
c      open(7,name='tv2.mat',status='new')
c      open(11,name='tv3.mat',status='new')
c      open(10,name='datam.dat',status='new')

c  Translate the Tv1 data

      do i=1,61
      read(8,*)p1
      x1(i)=p1
1020   enddo

      y1(1)=x1(1)
      y1(2)=x1(2)
      do i=3,59
         y1(i)=0.2*(x1(i-2)+x1(i-1)+x1(i)+x1(i+1)+x1(i+2))
      enddo
      y1(60)=x1(60)
      y1(61)=x1(61)

      do i=1,61
         write(9,1000)y1(i)
c         write(10,*) y1(i)
      enddo

1000   format(F10.2)

c  Translate the Tv2 data

      do i=1,61
      read(8,*)p1
      x1(i)=p1
1030   enddo

      y1(1)=x1(1)
      y1(2)=x1(2)
      do i=3,59
         y1(i)=0.2*(x1(i-2)+x1(i-1)+x1(i)+x1(i+1)+x1(i+2))
      enddo
      y1(60)=x1(60)
      y1(61)=x1(61)

      do i=1,61
         write(7,1010)y1(i)
         write(10,*) y1(i)
      enddo

1010   format(F10.2)
```


c Translate the Tv3 data

```
      do i=1,61
      read(8,*)p1
      x1(i)=p1
1040   enddo

      y1(1)=x1(1)
      y1(2)=x1(2)
      do i=3,59
         y1(i)=0.2*(x1(i-2)+x1(i-1)+x1(i)+x1(i+1)+x1(i+2))
      enddo
      y1(60)=x1(60)
      y1(61)=x1(61)

      do i=1,61
         write(11,1130)y1(i)
         write(10,*) y1(i)
      enddo

1130   format(F10.2)

      close(8)
      close(9)
      close(10)
      close(11)

      end
```

```
% This program PRO.M normalizes the smoothed data in TV.MAT and TM.MAT
% to the ambient value which is the first temp in the series of each
% and then converts the data from degrees F to degrees K.
% NOTE - THIS PROGRAM IS RUN AFTER THE TRANS.FOR PROGRAM IS RUN
```

```
load tv1
load tv2
load tv3
%
tvs1=tv1-tv1(1);
tvs1=tvs1*0.555;
%
tvs2=tv2-tv2(1);
tvs2=tvs2*0.555;
%
tvs3=tv3-tv3(1);
tvs3=tvs3*0.555;
%
datam=[tvs1' tvs2' tvs3']';
datam
save datam.dat datam /ascii
```

LIST OF REFERENCES

1. Reno, M. M., "Modeling Transfer Thermal Behavior in a Thrust Vector Control Jet Vane", Masters Thesis, Department of Mechanical Engineering, Naval Postgraduate School, Monterey, California, December 1992.
2. Nunn, R. H., "Jet Vane Modeling Development and Evaluation", Final Report from VRC Corporation, Monterey, California, January 1990.
3. Driels, M. R., "Heat Transfer Parametric Identification" Final Report FY92, Department of Mechanical Engineering, Naval Postgraduate School, Monterey, California, 1992.
4. Hatzenbuehler, M. A., "Modeling of Jet Vane Heat Transfer Characteristics and Simulation of Thermal Response", Masters Thesis, Department of Mechanical Engineering, Naval Postgraduate School, Monterey, California, June 1988.
5. Nunn, R. H. and Kelleher, M. D., "Jet Vane Heat Transfer Modeling", Research Report, Department of Mechanical Engineering, Naval Postgraduate School, Monterey, California, October 1986.

INITIAL DISTRIBUTION LIST

- | | | |
|----|---|---|
| 1. | Defense Technical Information Center
Cameron Station
Alexandria, Virginia 22304-6145 | 2 |
| 2. | Library, Code 52
Naval Postgraduate School
Monterey, California 93943-5002 | 2 |
| 3. | Department Chairman, Code ME
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, California 93943-5000 | 1 |
| 4. | Professor Morris R. Driels
Code ME/DR
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, California 93943-5000 | 2 |
| 5. | LT Gregory K. Parker
1144 Crestview St.
Reynoldsburg, Ohio 43068 | 2 |
| 6. | Naval Engineering Curricular Officer, Code 34
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, California 93943-5004 | 1 |
| 7. | A. Danielson
Thermal Structures Branch
Naval Air Weapons Center
Code C2891
China Lake, California 93555 | 2 |